

---

---

# Unicenter

## TCPaccess Communications Server Unprefixed Messages and Codes

Version 6.0



**Computer Associates**  
The Software That Manages eBusiness



This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies

# Contents

## Chapter 1: ABEND Codes

ABEND Codes .....	1-1
FTP Return Codes .....	1-33

## Chapter 2: API Sample Application Messages

Socket WHOIS .....	2-1
C WHOIS .....	2-3
C ACSHELLO .....	2-5
Socket ACSHELLO .....	2-9
FINGER .....	2-12
TTCP .....	2-13

## Chapter 3: API/IFS Internal Trace Entries

Flags and Return Codes .....	3-1
SRE Flags .....	3-1
TPL Flags .....	3-2
APICTFRE Return Codes .....	3-2
IFS Internal Trace Entries .....	3-3
ABND .....	3-3
CALL .....	3-3
CAP .....	3-6
DISP .....	3-7
DOWN .....	3-9
DRIV .....	3-9
EXIT .....	3-11
IUCV .....	3-12
LOCK .....	3-17
MSG .....	3-20
OE .....	3-20

---

PAGE .....	3-22
PMGR .....	3-24
POST .....	3-25
SAPI .....	3-25
SCHD .....	3-27
SSOB .....	3-28
STACK.....	3-28
TIME .....	3-31
TLI .....	3-32
TSO.....	3-35
UP .....	3-36
WAIT .....	3-36

## Chapter 4: API Return Codes

Returned Information.....	4-2
AOPEN/ACLOSE Macro Instructions .....	4-2
TPL-Based Macro Instructions .....	4-3
AOPEN and ACLOSE General Return Codes .....	4-4
AOPEN and ACLOSE Error Codes .....	4-6
TPL-Based General Return Codes – R15.....	4-11
Recovery Action Codes – R00.....	4-13
Conditional Completion Codes – R00.....	4-15
R00 Diagnostic Codes .....	4-16
Specific Error Codes.....	4-16
Recovery Action Code Combinations .....	4-17
Exceptional Conditions: RTNCD 04xx.....	4-18
Connection and Data Integrity Errors: RTNCD 08xx.....	4-19
Execution Environment Errors: RTNCD 0Cxx.....	4-20
Format or Specification Errors: RTNCD 10xx .....	4-22
Sequence and Procedural Errors: RTNCD 14xx.....	4-23
Logic Errors with No TPL Return Code: 18xx .....	4-25



---

## Chapter 5: API Diagnostic Codes

Diagnostic Codes.....	5-2
Converting Two-Byte to Four-Byte Diagnostic Codes.....	5-3
C006xxxx T010PNW.....	5-5
C100xxxx T010PPC.....	5-6
C101xxxx T010PSK.....	5-7
C102xxxx T010PAC.....	5-7
C107xxxx T010PSO.....	5-7
C200xxxx T010SPC.....	5-8
C201xxxx T010SSK.....	5-9
C202xxxx T010SAC.....	5-10
C203xxxx T010SBD.....	5-11
C204xxxx T010SCL.....	5-12
C205xxxx T010SCN.....	5-13
C206xxxx T010SGN.....	5-14
C207xxxx T010SGO.....	5-14
C208xxxx T010SSO.....	5-15
C209xxxx T010SIO.....	5-15
C20Axxxx T010SLI.....	5-16
C20Bxxxx T010SRD.....	5-17
C20Cxxxx T010SWR.....	5-18
C20Dxxxx T010SVR.....	5-20
C20Exxxx T010SVW.....	5-21
C20Fxxxx T010SRV.....	5-22
C210xxxx T010SSD.....	5-23
C211xxxx T010SFR.....	5-25
C212xxxx T010STO.....	5-26
C213xxxx T010SRM.....	5-27
C214xxxx T010SSM.....	5-28
C215xxxx T010SSL.....	5-30
C216xxxx T010SSN.....	5-31
C217xxxx T010SSH.....	5-31
C218xxxx T010SGH.....	5-31
C219xxxx T010SCA.....	5-32
C220xxxx T011SPC.....	5-32
C221xxxx T011SAC.....	5-33
C222xxxx T011SBD.....	5-36
C223xxxx T011SCA.....	5-39
C224xxxx T011SCL.....	5-42
C225xxxx T011SCN.....	5-45

---

C226xxxx T011SCP .....	5-48
C227xxxx T011SGH.....	5-51
C228xxxx T011SGN.....	5-54
C229xxxx T011SGO.....	5-57
C22Axxxx T011SGS.....	5-59
C22Bxxxx T011SID .....	5-62
C22Cxxxx T011SIN .....	5-65
C22Dxxxx T011SIO .....	5-69
C22Exxxx T011SLI.....	5-72
C22Fxxxx T011SRV .....	5-75
C230xxxxT011SSD.....	5-79
C231xxxxT011SSH.....	5-80
C232xxxxT011SSK.....	5-80
C233xxxxT011SSL .....	5-81
C234xxxxT011SSO.....	5-82
C235xxxxT011SSP .....	5-83
C236xxxxT011STS .....	5-83
C240xxxx (01xx) T012SPC .....	5-85
C241xxxx (03xx) T012TACC .....	5-86
C242xxxx (04xx) T012TADR .....	5-87
C243xxxx (05xx) T012TBIN.....	5-88
C244xxxx (06xx) T012TCLR.....	5-89
C245xxxx (07xx) T012TCLS.....	5-89
C246xxxx (08xx) T012TCNF.....	5-91
C247xxxx (09xx)T012TCON.....	5-92
C248xxxx (0Axx) T012TDIS .....	5-93
C249xxxx (0Bxx) T012TINF.....	5-93
C24Axxxx (0Cxx) T012TLIS .....	5-94
C24Bxxxx (0Dxx) T012TOPN.....	5-95
C24Cxxxx (0Exx) T012TOPT.....	5-97
C24Dxxxx (0Fxx) T012TRCV .....	5-99
C24Exxxx (10xx) T012TRER.....	5-100
C24Fxxxx (11xx) T012TRFR .....	5-101
C250xxxx (12xx) T012TRJT .....	5-102
C251xxxx (13xx) T012TRLK.....	5-103
C252xxxx (14xx) T012TRLS.....	5-104
C253xxxx (15xx) T012TRCT .....	5-104
C254xxxx (16xx) T012TSND .....	5-105
C255xxxx (17xx) T012TSTO .....	5-106
C256xxxx (18xx) T012TUNB.....	5-107
C257xxxx (19xx) T012TUSR.....	5-108

---

C258xxxx (1Axx) T012AOPN.....	5-109
C259xxxx (1Bxx) T012ACLS.....	5-110
C25Axxxx (1Cxx) T012TCHK.....	5-111
C25Bxxxx (1Dxx) T012TERR.....	5-112
C25Cxxxx (1Exx) T012TSTA.....	5-112
C25Fxxxx (21xx) T01PSTUB.....	5-113
C262xxxx (6Fxx) T012TPLK.....	5-114
C301xxxx (24xx) T01XACPT.....	5-116
C302xxxx (25xx) T01XBIND.....	5-116
C303xxxx (26xx) T01XCLS1.....	5-117
C304xxxx (27xx) T01XCLS2.....	5-117
C305xxxx (28xx) T01XCONN.....	5-118
C306xxxx (29xx) T01XUBND.....	5-119
C307xxxx (2Axx) T01XCREA.....	5-119
C308xxxx (2Bxx) T01XFREE.....	5-120
C30Cxxxx (2Dxx) T01XLIST.....	5-120
C30Dxxxx (2Exx) T01XOPT1.....	5-121
C30Exxxx (2Fxx) T01XPEER.....	5-123
C30Fxxxx (30xx) T01XPRTA.....	5-123
C311xxxx (31xx) T01XREAD.....	5-124
C312xxx T01XOPT2.....	5-125
C315xxxx (33xx) T01XSHT1.....	5-127
C316xxxx (34xx) T01XSHT2.....	5-127
C317xxxx (35xx) T01XSND1.....	5-128
C318xxxx (36xx) T01XSND2.....	5-129
C404xxxx (3Bxx) T01ASWDN.....	5-129
C4FFxxxx (37xx) T01ASFRR.....	5-129
C502xxxxT01ESCF.....	5-129
C506xxxxT01ESDI.....	5-130
C508xxxxT01ESTP.....	5-130
C512xxxxT01EUCF.....	5-130
C516xxxxT01EUDI.....	5-130
C518xxxxT01EUTP.....	5-131
C526xxxx (43xx) T01ETDI.....	5-131
C528xxxx (45xx) T01ETTP.....	5-131
C602xxxx T01SMOVE.....	5-132
C704xxxx T01AMIUC.....	5-132
C805xxxx T01XTTCP.....	5-133
C901xxxx T01SIMUX.....	5-133
C909xxx T01SISND.....	5-134
C920xxx T01SUBN.....	5-134

---

C921xxx T01SUIN .....	5-135
C922xxx T01SUSND .....	5-135
C923xxxx T01SUUNB .....	5-135
C930xxxx T01SRBND .....	5-136
C931xxx T01SRIN.....	5-136
C932xxxx (5Bxx) T01SRSND.....	5-136
C933xxxx (5Cxx) T01SRUNB.....	5-137
C940xxxx (5Dxx) T01STCLO .....	5-137
C941xxxx (5Exx) T01STCON .....	5-138
C942xxxx (5Fxx) T01STFIN.....	5-138
C943xxxx (60xx) T01STIMU.....	5-139
C944xxxx (61xx) T01STIN .....	5-139
C945xxxx (62xx) T01STLIS.....	5-140
C946xxxx (63xx) T01STREA.....	5-140
C947xxxx (64xx) T01STRST.....	5-141
C948xxxx (65xx) T01STSND .....	5-141
C94Cxxxx (69xx) T01STTMK.....	5-142
C94Dxxxx (6Axx) T01STTML.....	5-143
C94Exxxx (6Bxx) T01STTMP .....	5-143
C94Fxxxx (6Cxx) T01STTMR.....	5-144
C950xxxx (6Dxx) T01STTMT .....	5-145
C951xxxx (6Exx) T01STUNB .....	5-145
Ilatch Errors.....	5-146
DGMAKE/AHAP Log Errors .....	5-146
Security Access Errors.....	5-147

## Chapter 6: C Socket errno Codes

errno Code Table.....	6-1
IPRCODE Table.....	6-5

---

## Chapter 7: DNR Return Codes

General Return Codes.....	7-2
Recovery Action Codes .....	7-3
Conditional Completion Codes: RTNCD 00xx.....	7-5
Specific Error Codes: xxxx.....	7-6
Valid Combinations .....	7-6
Exceptional Conditions: RTNCD 04xx .....	7-7
Execution Environment Errors: RTNCD 08xx.....	7-16
Format or Specification Errors: RTNCD 0Cxx.....	7-24
Sequence and Procedural Errors: RTNCD 10xx.....	7-28
Logic Errors with No DPL Return Code: RTNCD 14xx.....	7-29

## Chapter 8: FTP Server Messages

FTP Server Messages .....	8-1
---------------------------	-----

## Chapter 9: MAIL Messages

SNDSMSG Messages .....	9-1
SSMTP Messages .....	9-2

## Chapter 10: Error Messages

Network Error Messages.....	10-2
Disk perror Messages.....	10-5
Other Error Messages .....	10-9
Mail Error Messages .....	10-9
User Configuration or Subsystem Error Messages.....	10-15
Socket Library perror Messages .....	10-27

## Chapter 11: RPCINFO Messages

Message Format.....	11-1
RPCINFO Messages .....	11-2

---

## Chapter 12: RPC/XDR Errors

RPC Library Error Messages .....	12-1
Message Formatting.....	12-8

## Chapter 13: Telnet Messages

Server Telnet Messages .....	13-1
------------------------------	------

# ABEND Codes

---

This chapter lists ABEND codes.

The following types of ABENDs can occur:

- ABEND – Standard OS User ABENDs
- IABEND – ABENDs forced by the ICT on a PTASK
- PABEND – Pseudo ABENDs initiated by a PTASK

This chapter includes:

- [ABEND Codes](#) – ABEND codes 001 through F11
- [FTP Return Codes](#) – Return codes set by the FTP and FTP2 programs when they terminate

In this chapter, ABENDs are identified with the ABEND number in hexadecimal followed by an Uxxx number, where xxx is the OS User ABEND code in decimal.

## ABEND Codes

These codes are presented in alphanumeric order from 001 through F11.

### IABEND 001 (U1)

Reason:	The commutator is validating an IECB or ILIST but the IECB or ILIST is invalid.
Action:	Save all output from the job. Check the WTO and job logs for any related messages. If the user is using P-services to create an application, check the program logic of the failing PTASK. Check the parameter lists passed to the P-service. If the user is using the default applications, contact Customer Support.

### IABEND 002 (U2)

- Reason: The commutator is validating an ECB for use by the PPOST facility but finds the ECB is not valid, usually not fullword aligned. This ABEND happens through a program exception.
- Action: Save all output from the job. Check the WTO and job logs for any related messages. If the user is using P-services to create an application, check the program logic of the failing PTASK. Check the parameter lists passed to the P-service. If the user is using the default applications, contact Customer Support.

### IABEND 003 (U3)

- Reason: The commutator's I/O services routine was called with an invalid entry code.
- Action: Collect all job output from the job including maps and dumps, and contact Customer Support.

### ABEND 008 (U8)

- Reason: The PATTACH P-service is called with a bad parameter list or PTA address. The calling PTASK ABENDs.
- Action: Save all output from the job. Contact Customer Support.
- If you are:
- Writing applications using P-services, check your PATTACH parameter list
  - Using the default applications, contact Customer Support

### ABEND 009 (U9)

- Reason: The PSTATUS P-service is called with an invalid entry code in register R0. The entry code is larger than the maximum allowable entry code. The calling PTASK ABENDs.
- Action: Save all output from the job. Check the WTO and job logs for related messages.
- If you are:
- Writing applications using P-services, check the logic of your PSTATUS calls and parameter lists
  - Using the default applications, contact Customer Support



**ABEND 00B (U11)**

Reason: The PSNAP P-service is called with a bad parameter list. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PSNAP parameter list
- Using the default applications, contact Customer Support

**ABEND 00E (U14)**

Reason: The PDEBUG P-service is called with a bad parameter list. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PDEBUG parameter list
- Using the default applications, contact Customer Support

**ABEND 012 (U18)**

Reason: The PSTOP P-service is called with a bad parameter list. The service code in register R0 contains a number greater than two. The only valid register 0 values are 0, 1, and 2. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PSTOP parameter list
- Using the default applications, contact Customer Support

### ABEND 013 (U19)

Reason: PATTN, the install/remove IOS attention handler service, is called with an invalid DCB address or unopened DCB. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PATTN parameter list
- Using the default applications, contact Customer Support

### ABEND 014 (U20)

Reason: PDMIH P-service to disable the missing attention handler is called with an invalid DCB address or unopened DCB. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PDMIH parameter list
- Using the default applications, contact Customer Support

### ABEND 015 (U21)

Reason: The PDRVER P-service is called with an invalid DCB address or unopened DCB. The PDRVER P-service can do these functions:

- Enable or disable the missing interrupt handler
- Initialize sense byte count
- Add address of the channel appendage to the DEB

The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PDRVER parameter list.
- Using the default applications, contact Customer Support

**IABEND 016**

Reason: The PPDSENQ P-service was called with a bad parameter list address. This may generate unexpected program checks. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PPDSENQ parameter list
- Using the default applications, contact Customer Support

**PABEND 080 (U128)**

Reason: The stdin and stdout connections will not open. The VTAM connection cannot be opened for some reason. A PABEND of x'80' abends the PTASK.

Action: Check to make sure the Unicenter TCPaccess VTAM APPLs were defined to VTAM. Check to make sure you are using the correct ACBNAME parameter in the APPCFGxx member.

**PABEND 100 (U256)**

Reason: The stderr connection will not open. Unicenter TCPaccess error log cannot be written. A PABEND of x'100' abends the PTASK.

Action: Make sure the Unicenter TCPaccess job has all its WTOLOGxx DD statements.

**IABEND 101 (U257)**

Reason: The commutator is validating an ECB or ECBLIST but the ECB or ELIST is invalid.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If the user is using P-services to create an application, check the program logic of the failing PTASK. Check the parameter lists passed to the P-service. If the user is using the default applications, contact Customer Support.

### IABEND 102 (U258)

Reason: The commutator finds that the PSTA INDEX or PTA address for an ECB is not valid. This ABEND happens through a program exception.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

### ABEND 104 (U260)

Reason: A CALL is made to PCOREF to *free* PCORE storage, but the storage is not owned by PRSMGR.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

### PABEND 107 (U263)

Reason: A PTASK calls the PACCESS P-service with a bad parameter list. The PTASK ABENDs with a X'107'.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

**ABEND 108 (U264)**

Reason: The PATTACH TYPE=ADD P-service is called with a PTA address of the parent PTASK. The PTA address of the parent PTASK is actually a descendant of the PTASK that issued the PATTACH TYPE=ADD. This is a logic error by the programmer (for if the parent PTASK ends, there is no PTASK to post). The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check your logic
- Using the default applications, contact Customer Support

**ABEND 109 (U265)**

Reason: The PSTATUS NEXT P-service is called out of sequence. The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check the logic of your PSTATUS calls and parameter lists
- Using the default applications, contact Customer Support

**ABEND 10D (U269)**

Reason: A PRSMGR DELETE P-service call is made. Module PRSMGRD is called to delete an RE# element. The RE# address is not found. The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check the logic of your PRSMGR calls and parameter lists
- Using the default applications, contact Customer Support

### IABEND 10E (U270)

Reason: The PDEBUG P-service is called with a bad request type in the parameter list. The request type could be either an undefined type or a request to delete or replace a PDEBUG intercept that does not exist. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PDEBUG calls for a correct request type in the PDEBUG parameter list
- Using the default applications, contact Customer Support

### ABEND 110 (U272)

Reason: A call is made to EXTGIVE to *give* an EXCH window, but the PTA address is invalid. This ABEND is a result of a PEXCH P-service call. The PEXCH P-service call is valid only in an MVT environment. Application developers should not be using the PEXCH P-service. This ABEND should never occur.

Action: Save all output from the job. Contact Customer Support.

### ABEND 111 (U273)

Reason: ACDYNAL is called to open or close a list of ACBs or DCBs but lists are not allowed.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

### ABEND 112 (U274)

Reason: PSTOP is called for set application stop ECB address. The ECB address is invalid. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support. If you are writing applications using P-services, check the ECB address of the calling PSTOP parameter list.

**ABEND 113 (U275)**

Reason: PATTN, the install/remove IOS attention handler service, is called. The CVTJESCT in the CVT does not point to a JES2 or JES3. Unicenter TCPaccess cannot find an unused attention index. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

**IABEND 116**

Reason: The PPDSENQ P-service was called to enqueue a resource; however, after completion of the process the dequeue failed and the resource remained enqueued. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PPDSENQ parameter list
- Using the default applications, contact Customer Support

**ABEND 142 (U322)**

Reason: A PTASK has consumed more than the maximum allowable CPU time. Unicenter TCPaccess believes the PTASK is looping, terminates the PTASK and dumps related storage to the PTASK.

Action: Save all output from the job. A dump of a related PTASK before the 142 PTASK ABEND may be related to this looping problem. Check the WTO and job logs for any related messages. Contact Customer Support.

**IABEND 16D (U365)**

Reason: ACHSM is called to recall a migrated data set, but the PDYNAL parameter list is invalid.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-services
- The default applications, contact Customer Support

### **PABEND 180 (U384)**

Reason: The TCP connection will not open. ATOPN failed for this process. A PABEND of x'180' abends the PTASK.

Action: Check that the other host that Unicenter TCPaccess is trying to connect to is up on the network. If the remote host is up, save all output from the job. Contact Customer Support.

### **PABEND 200 (U512)**

Reason: The program requests an amount of storage through PCORE. The storage is not available. The PTASK decides to ABEND rather than wait. The PTASK ABENDs with an x'200'.

Action: Retry the task. Save all output from the job.

### **PABEND 201 (UC9)**

Reason: A PTASK issued a PWAIT call while still having an active PFRR set.

Action: Contact Customer Support.

### **ABEND 204 (U516)**

Reason: A call is made to PC\$GIVE to *give* PCORE storage, but the PTA is invalid.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support.



**PABEND 207 (U519)**

Reason: A PTASK calls the PACCESS TYPE=LOGOFF P-service trying to logoff, yet the PTASK is not logged on to Unicenter TCPaccess. A PABEND of x'207' occurs.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

**ABEND 209 (U521)**

Reason: During processing of a PSTATUS P-service, an invalid PTA address is returned from ICKPTA. The PSTATUS P-service cannot continue. The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check the logic of your PSTATUS calls and parameter lists
- Using the default applications, contact Customer Support

**ABEND 20D (U525)**

Reason: Either the PRSMGR NEXT or PRSMGR DLTCUR P-service is called in an invalid sequence. The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check the logic of your PRSMGR calls and parameter lists
- Using the default applications, contact Customer Support

### **ABEND 210 (U528)**

Reason: A call is made to EX\$FND to find a RE resource for a window, but the window ID is not known. This ABEND is a result of a PEXCH P-service call. The PEXCH P-service call is valid only in an MVT environment. Application developers should not be using the PEXCH P-service. This ABEND should never occur.

Action: Save all output from the job. Contact Customer Support.

### **ABEND 211 (U529)**

Reason: ACDYNAL is called with an invalid parameter list.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support.

### **ABEND 212 (U530)**

Reason: The PSTOP P-service is processing a PSTOP exit request when a logic error is uncovered that prevents further processing. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

### **ABEND 213 (U531)**

Reason: PATTN, the install/remove IOS attention handler service, is called. The attention index that Unicenter TCPaccess was going to use is reserved or invalid. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

**IABEND 216**

Reason: The PPDSENQ P-service was called to enqueue a resource; however, after completion of the process the dequeue failed and the resource remained enqueued. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PPDSENQ parameter list.
- Using the default applications, contact Customer Support

**IABEND 26D (U621)**

Reason: ACHSM is called to recall a migrated data set, but the conditional GETMAIN fails to obtain storage.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-services.
- The default applications, contact Customer Support

**PABEND 307 (U775)**

Reason: A PTASK calls the PACCESS TYPE=LOGOFF P-service trying to logoff; the ACEE pointer is invalid or the logoff fails. A PABEND of x'307' occurs.

Action: Save all output from the job. Check the WTO and logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

### **ABEND 309 (U777)**

Reason: During processing of a PSTATUS STOP P-service, 250 related PTASKs are found. The maximum number of descendants that PSTATUS STOP processes is 249. The calling PTASK ABENDs.

Action: Save all output from the job. Check the WTO and job logs for related messages. If you are writing applications using P-services, check the logic of your PSTATUS calls and parameter lists. If you are using the default applications, contact Customer Support.

### **ABEND 311 (U785)**

Reason: ACDYNAL is attempting to find a RE# resource element for an ACB or DCB but none is found.

Action: Save all output from the job. Contact Customer Support.

### **ABEND 313 (U787)**

Reason: PATTN, the install/remove IOS attention handler service, is called. A PC call to update the UCB or ATTN fails. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

### **ABEND 314 (U788)**

Reason: PDMIH, the disable/enable missing interrupt handler service, is called. A PC call to update the UCB fails. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

### **ABEND 315 (U789)**

Reason: The PDRVER is called. The PDRVER P-service can perform these functions:

- Enable or disable the missing interrupt handler
- Initialize sense byte count
- Add address of the channel appendage to the DEB

A PC routine fails to update the UCB and DEB control blocks. The calling PTASK ABENDs.

Action: Save all output from the job. Contact Customer Support.

**IABEND 316**

Reason: The PPDSENQ P-service was called to enqueue a resource, but the corresponding flag in the PPDSENQ parameter list was not set up correctly. This is detected as a logic error. The calling PTASK ABENDs.

Action: Save all output from the job.

If you are:

- Writing applications using P-services, check your PPDSENQ parameter list
- Using the default applications, contact Customer Support

**IABEND 36D (U877)**

Reason: ACHSM is called to recall a migrated data set, but OS attach for the recall task fails.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Verify the recall task ACRECALL is link edited within ACCOM. If the problem persists, contact Customer Support.

**ABEND 3E8 (U1000)**

Reason: A PTASK tries to perform a P-service while on the BYPASS task. PTASKs can issue P-services only while on the commutator task. The PTASK is ABENDED with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK
- The default applications, contact Customer Support

### **ABEND 3F2 (U1010)**

- Reason: A PTASK calls program IQCELLG to allocate quick cell memory. Not enough memory exists for the request. A conditional GETMAIN is issued by Unicenter TCPaccess for more storage. Less than 72 bytes of storage are returned from the GETMAIN. Unicenter TCPaccess is out of memory for the PTASK. No storage is available from quick cell storage. The PTASK terminates with a dump.
- Action: Save all output from the job. Examine the storage in the dump for a proliferation of any type of storage throughout memory. Check the WTO and job logs for any related messages. Raise the region parameter on the Unicenter TCPaccess startup JCL step. If the problem persists, contact Customer Support.

### **PABEND 400 (U1024)**

- Reason: The input initialization parameters are invalid. The program cannot execute any further. An ABEND of x'400' causes the PTASK to ABEND.
- Action: Check APPCFGxx input. Contact Customer Support.

### **ABEND 403 (U1027)**

- Reason: During startup for the TCP base product, an attempt was made to perform a SETLOCK OBTAIN for the LOCAL lock, but failed. Startup for Unicenter TCPaccess terminates.
- Action: Save the output of the job. Contact Customer Support.

### **ABEND 404 (U1028)**

- Reason: During startup for the TCP base product, an attempt was made to GETMAIN SUBPOOL 241 Key 0 CSA storage, but failed. Startup for the TCP base product is then terminated.
- Action: Save the output of the job. Contact Customer Support.

### **ABEND 405 (U1029)**

- Reason: During startup for the TCP base product, an attempt was made to perform a SETLOCK RELEASE for the LOCAL lock, but failed. Startup terminates.
- Action: Save the output of the job. Contact Customer Support.

**IABEND 406 (U1030)**

- Reason: ACLOAD attempted to load a module but failed due to insufficient virtual storage.
- Action: Increase the virtual storage available to the Unicenter TCPaccess address space and restart Unicenter TCPaccess.

**IABEND 411 (U1041)**

- Reason: ACDYNAL is called to allocate a migrated data set, but the PDYNAL parameter list is invalid.
- Action: Save all output from the job. Check the WTO and logs for any related messages.
- If you are using:
- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-services
  - The default applications, contact Customer Support

**IABEND 416**

- Reason: The P-service FREE RESOURCE routine failed to dequeue an enqueued resource. The calling PTASK ABENDs.
- Action: Save all output from the job.
- If you are:
- Writing applications using P-services, check your PPDSNQ parameter list
  - Using the default applications, contact Customer Support

**ABEND 444 (U1092)**

- Reason: Program SPOOL#4 found an error condition that normally would cause SPOOL#4 to terminate without a dump; however ACTEST command SPOOLDMP ON was entered. When SPOOLDMP is ON, program SPOOL#4 terminates ptask SPOOL#4 and dumps under certain error conditions (having to do with open, close, dynamic allocation, and so forth.).
- Action: Generally you should only use SPOOLDMP ON to debug other problems within SPOOL#4 at the direction of Customer Support.

### ABEND 44C (U1100)

Reason: A PTASK tries to perform a P-service with an invalid PTA address in register R11. By convention, R11 should contain the caller's PTA address. The PTASK ABENDs with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK
- The default applications, contact Customer Support

### ABEND 456 (U1110)

Reason: A PTASK calls program IQCELLG to allocate quick cell memory. The storage amount requested is either greater than x'FFFFFF' or zero. A bad parameter was passed. The PTASK terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK
- The default applications, contact Customer Support

### ABEND 457 (U1111)

Reason: The quick cell get logic detected that the next pointer in the chain of free quick cells was corrupted. An attempt is made to salvage the chain, then the currently executing PTASK abends.

Action: This indicates an internal error within Unicenter TCPaccess and requires the SVC dump output to diagnose.



**ABEND 458 (U1112)**

- Reason: The quick cell get logic detected that the next pointer in the chain of free quick cells was corrupted. An attempt is made to salvage the chain, then the currently executing PTASK abends.
- Action: This indicates an internal error within Unicenter TCPaccess and requires the SVC dump output to diagnose.

**ABEND 459 (U1113)**

- Reason: The quick cell get logic detected that the next pointer in the chain of free quick cells was corrupted. An attempt is made to salvage the chain, then the currently executing PTASK abends.
- Action: This indicates an internal error within Unicenter TCPaccess and requires the SVC dump output to diagnose.

**ABEND 45A (U1114)**

- Reason: The quick cell get logic detected that the next pointer in the chain of free quick cells was corrupted. An attempt is made to salvage the chain, then the currently executing PTASK abends.
- Action: This indicates an internal error within Unicenter TCPaccess and requires the SVC dump output to diagnose.

**ABEND 4B0-4D8 (U1200-1240)**

Reason: ABEND codes in the range 1200 to 1240 are issued by the SAS/C runtime routines. This table provides a brief summary:

Code	Description
1200	Corruption of auto storage control blocks while DSA was being freed
1201	Corruption of auto storage control blocks during new DSA allocation
1202	During program termination, program stored past end of automatic storage stack
1203	Corruption of auto storage control blocks while freeing automatic storage
1204	Target of a longjmp cannot be found
1205	Corruption of heap storage control blocks during calloc/malloc
1206	Corruption of heap storage control blocks when calloc/malloc tried to reuse portion of heap
1207	Corruption of heap storage control blocks detected by free
1208	Free detected invalid pointer to a block
1209	Corruption of head storage control blocks during program termination
1210	SIGABRT is raised and default handling in effect
1211	Invalid argument to unloadm
1212	Required transient runtime functions cannot be loaded
1213	Subordinate C load module is called directly
1214	More than one recursive error occurred during diagnostic message processing
1215	(CMS) tried to delete storage occupied by dynamically loaded module
1216	(CMS) invalid library control block during call to unloadm
1217	(CMS) invalid library control block during call to loadm
1218	(CMS) error trying to delete a buffer during dynamic loading
1219	Error in interface between debugger and library
1220	Abort debugger command executed

Code	Description
1221	(CMS) CP detected error loading a library segment
1222	Library framework manager detected an error
1223	Indicates system problem in TSO SUBCOM interface
1224	longjmp or exit attempted to terminate a non-C routine
1225	SIGALARM or SIGIUCV lacks a handler
1226	Full-Screen Support library detected an internal error
1227	Library error occurred, terminating a process
1228	Insufficient storage to extend the internal queue of IUCV messages
1233	Internal error in inter-language communication support routines
1234	Call was made from C program to another high-level language but framework not created
1235	Inter-language communication feature usage error
1240	Unexpected program check during library ABEND analysis

Action: Refer to your SAS/C Documentation for further information or call Customer Support.

### ABEND 4E2 (U1250)

Reason: A PTASK calls program IQCELLF to release quick cell memory. The storage address is not on the list of allocated quick cell pools. The PTASK terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK
- Using the default applications, contact Customer Support

### **IABEND 511 (U1297)**

- Reason: ACDYNAL is called to allocate a migrated data set, but ACQCELL get fails to get storage for the HSM recall.
- Action: Save all output from the job. Check the WTO and job logs for any related messages. Check the region size of Unicenter TCPaccess . If the problem persists, contact Customer Support.

### **ABEND 51E (U1310)**

- Reason: A PTASK calls program IQCELLF to release quick cell memory. The storage amount to be released does not end on a double word boundary. The PTASK terminates with a dump.
- Action: Save all output from the job. Check the WTO and job logs for any related messages.
- If you are using:
- P-services to create an application, check program logic of the failing PTASK
  - Using the default applications, contact Customer Support

### **ABEND 582 (U1410)**

- Reason: PTASK calls program IQCELLF to release quick cell memory. The storage amount to be released is either larger than x'FFFFFF' or zero. Bad parameters were passed to IQCELLF. The PTASK terminates with a dump.
- Action: Save all output from the job. Check the WTO and job logs for any related messages.
- If you are using:
- P-services to create an application, check the program logic of the failing PTASK
  - Using the default applications, contact Customer Support

### **ABEND 706 (U1798)**

- Reason: ACLOAD attempts to load a module that is identified as non-executable.
- Action: Determine the problem with the module and restart Unicenter TCPaccess .

**ABEND 7CF (U1999)**

- Reason: A PTASK issues the XPOST P-service and not enough storage is available from IQCELLG to allocate an XAE for the service. Unicenter TCPaccess is out of storage. The PTASK terminates with a dump.
- Action: Save all output from the job. Check storage for a proliferation of one type of storage to see if there might be some kind of storage allocation loop. Check the WTO and job logs for any related messages. If the problem persists, contact Customer Support.

**ABEND 7D0 (U2000)**

- Reason: The commutator is attempting to build a list of ECBs from PTA ECB lists, but not enough stack storage is available to hold the entire list.
- Action: Save all output from the job. Check the WTO and job logs for any related messages.
- If the user is using:
- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
  - The default applications, contact Customer Support

**ABEND 7D3 (U2003)**

- Reason: Subroutine IERRWAIT detects all PTASKs in an error wait condition. No work can be performed. Unicenter TCPaccess terminates.
- Action: Save all output from the job. U2003 ABENDs indicate some function within Unicenter TCPaccess has repeatedly failed. Check the WTO and job logs for related messages. Contact Customer Support.

**ABEND 7D7 (U2007)**

- Reason: A PTASK is attempting to call the ACSECPC routine when it discovers that the ACSSCB pointer is zero. The ACSSCB pointer has been overwritten. DORACFPC subroutine discovers this problem. Unicenter TCPaccess can no longer provide an external security interface. An ABEND of x'7D7' occurs. Unicenter TCPaccess terminates from this ABEND.
- Action: Save all output from the job. Contact Customer Support.

### **ABEND 7DD (U2013)**

- Reason: Routine IQCELLG is called to allocate a RE#. No storage is available from quick cell storage. The PRSMGR routine cannot continue. Unicenter TCPaccess terminates with a dump.
- Action: Save all output from the job. Check the WTO and job logs for related messages. If there is a proliferation of the same kind of storage, a PTASK may be allocating storage in a loop.
- Retry the operation first. If it fails again, bring Unicenter TCPaccess down. Raise the region size on the Unicenter TCPaccess job. Start up Unicenter TCPaccess . Retry the operation again. If the error recurs, contact Customer Support.

### **ABEND 7E2 (U2018)**

- Reason: Two conditions can cause a U2018 to occur: ACDEBUG's SPIE exit is called with an invalid PTA address or the Bypass task ABENDs. Unicenter TCPaccess terminates with a dump.
- Action: Restart the Unicenter TCPaccess job. Save all output from the job. Check Unicenter TCPaccess WTO and job logs for related information. Contact Customer Support.

### **ABEND 7EE (U2030)**

- Reason: Not enough initial core is available to build the Unicenter TCPaccess control blocks. Unicenter TCPaccess terminates with a dump.
- Action: Increase the REGION parameter on the Unicenter TCPaccess job step. Restart the Unicenter TCPaccess job. If the problem persists, contact Customer Support.

### **ABEND 7EF (U2031)**

- Reason: Unicenter TCPaccess cannot successfully open the DCB for either the ACSNAP or the ACLOG DD statement. The ACLOG DD is the ICT logging file. The ACSNAP DD contains formatted dumps of any ABENDING PTASKs. Unicenter TCPaccess cannot continue and terminates with a dump.
- Action: Check the Unicenter TCPaccess startup JCL for ACSNAP and ACLOG DD statements.

**ABEND 800 (U2048)**

Reason: A serious error occurred (an ABEND of x'800'), causing Unicenter TCPaccess to terminate. Unicenter TCPaccess control blocks have been corrupted.

Action: Save all output from the job. Contact Customer Support.

**ABEND 834 (U2100)**

Reason: The commutator is attempting to ACALL the BYPASS tasks but is unsuccessful because the current TCB is not the BYPASS task TCB.

Action: Save all output from the job. Check the WTO and job logs for any related messages. If the user is using P-services to create an application, check the program logic of the failing PTASK. Check the parameter lists passed to the P-service. If the user is using the default applications, contact Customer Support.

**ABEND 837 (U2103)**

Reason: During termination processing, subroutine TSK\$DEL discovers PTASKs in an error wait condition. Unicenter TCPaccess terminates and a dump is taken.

Action: Save all output from the job. U2103 ABENDs indicate some function within Unicenter TCPaccess has failed. Check the Unicenter TCPaccess WTO and job logs for related messages.

**ABEND 83B (U2107)**

Reason: A PTASK attempting to call the ACSECPC routine discovers that the ACSSCB pointer is either invalid or has been violated by another program. DORACFPC subroutine discovered this problem. Unicenter TCPaccess can no longer provide an external security interface. An ABEND of x'83B' occurs. Unicenter TCPaccess terminates from this ABEND.

Action: Save all output from the job. Contact Customer Support.

**ABEND 846 (U2118)**

Reason: No RE# element can be allocated in ACDEBUG's STAE exit. No storage is available from quick cell storage. The STAE exit cannot continue. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for related messages. If there is a proliferation of the same kind of storage, a PTASK may be allocating storage in a loop.

Retry the operation first. If it fails again, bring Unicenter TCPaccess down and increase the region size on the Unicenter TCPaccess job. Restart Unicenter TCPaccess and retry the operation. If the error recurs, contact Customer Support.

### **ABEND 852 (U2130)**

**Reason:** The save area passed to module ACCONFG is not the first save area. ACCONFG is an initialization module. No PTASK should ever call module ACCONFG. Unicenter TCPaccess terminates with a dump.

**Action:** Save output from the job. Check the WTO and job logs for any related messages.

If you are using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

If the problem occurs after startup, submit the Unicenter TCPaccess job. Check the logs for related messages. If the problem persists, contact Customer Support.

### **ABEND 853 (U2131)**

**Reason:** Initialization and termination module ACCALL tries to allocate quick cell storage from IQCELLG. No storage is available from quick cell storage. Unicenter TCPaccess terminates with a dump.

**Action:** Restart the Unicenter TCPaccess job. Save all output from the Unicenter TCPaccess job. Examine the storage in the dump for a proliferation of any type of storage throughout memory. Check the WTO and job logs for any related messages. If the failure occurs during Unicenter TCPaccess initialization, raise the region parameter on the Unicenter TCPaccess start-up JCL step. If the problem persists, contact Customer Support.

### **ABEND 8AA (U2218)**

**Reason:** No DB#A element can be allocated in GETPDA for ACDEBUG's STAE, SPIE PABEND, or DB\$FREE routines. No storage is available from quick cell storage. The exit cannot continue. Unicenter TCPaccess terminates with a dump.

**Action:** Restart the Unicenter TCPaccess job. Save all output from the Unicenter TCPaccess job. Examine the storage in the dump for a proliferation of any type of storage throughout memory. Check the WTO and job logs for any related messages. Contact Customer Support.



**ABEND 8B7 (U2231)**

Reason: Initialization module ACCALL tries to IDENTIFY the IOSSUB routine but fails. Unicenter TCPaccess terminates with a dump.

Action: Restart the Unicenter TCPaccess job. Save all output from the Unicenter TCPaccess job. Check the WTO and job logs for any related messages. If the problem persists, contact Customer Support.

**ABEND 906 (U2310)**

Reason: A module load use count overflows.

Action: Determine if module was created correctly.

**ABEND 90E (U2318)**

Reason: A recursive call to Unicenter TCPaccess SPIE exit is made. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the Unicenter TCPaccess job. Check the SYSABEND DD for any related dump. Check the WTO and job logs for any related messages. Restart the Unicenter TCPaccess job. Contact Customer Support.

**ABEND 91B (U2331)**

Reason: Initialization module ACCALL issues an ATTACH for the commutator task. The ATTACH fails. Unicenter TCPaccess cannot continue without a commutator task. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Examine the dump. Restart the Unicenter TCPaccess job. If the problem persists, contact Customer Support.

**ABEND 972 (U2418)**

Reason: A recursive call to Unicenter TCPaccess STAE exit is made. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Restart the Unicenter TCPaccess job. Contact Customer Support.

### **ABEND 9D6 (U2518)**

Reason: Unicenter TCPaccess is looping. No PTASK seems to be active on the commutator. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Restart the Unicenter TCPaccess job. Contact Customer Support.

### **ABEND A3A (U2618)**

Reason: On return from PDEBUG's exception exit, the RE# element cannot be located. Processing cannot continue. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Restart the Unicenter TCPaccess job. Contact Customer Support.

### **ABEND A9E (U2718)**

Reason: Unicenter TCPaccess STAE initialization call fails. Processing can no longer continue. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Restart the Unicenter TCPaccess job. Contact Customer Support.

### **ABEND B02 (U2818)**

Reason: A program exception occurs on the commutator task. Unicenter TCPaccess finds that its IRB chain has been corrupted. Processing can no longer continue. Unicenter TCPaccess terminates with a dump.

Action: Save all output from the job. Check the WTO and job logs for any related messages. The previous ABEND may have corrupted the IRB chain. Restart the Unicenter TCPaccess job. Contact Customer Support.

### **ABEND B0F (U2831)**

Reason: The commutator encounters a stack overflow.

Action: Contact Customer Support.

**ABEND B66 (U2918)**

- Reason: A critical Unicenter TCPaccess PTASK ABENDs. Processing can no longer continue. Unicenter TCPaccess terminates with a dump.
- Action: Restart the Unicenter TCPaccess job. Save all output from the job. Check the WTO and job logs for any related messages. Contact Customer Support.

**AABEND BB7 (U2999)**

- Reason: An ACTEST user issues the ABEND command. Unicenter TCPaccess terminates with a dump.
- Action: None, if the ACTEST user intended to terminate Unicenter TCPaccess. If ACTEST is not protected by command security and a curious user entered the command (and terminated Unicenter TCPaccess), the system programmer responsible for Unicenter TCPaccess might consider implementing command security for ACTEST.

**ABEND C06 (U3078)**

- Reason: ACLOAD attempts to load a module but the module is found to contain invalid attributes.
- Action: The module must be generated without an overlay structure or similar attribute.

**ABEND C64 (U3172)**

- Reason: The driver issues this user ABEND under a variety of severe error conditions. This table lists the reason codes and descriptions:

Code	Description
1	The DIE exit was unable to translate the real CCW address to a virtual CCW address, because either the end of the LNIO chain was reached or the command chaining bit in LNIOCCW1 was not set.
2	During DIE processing, an LRA (load real address) instruction returned a non-zero return code, indicating that the page was not page-fixed. This can be caused by either the page being page-freed prematurely or a logic error in this program not using the correct value with the LRA.
3	After determining which LNIO was the last to complete, a check indicated that there was another LNIO on the chain.

Code	Description
	This indicates the channel program was not built correctly.
4	An attempt was made to enqueue an I/O request on the ready queue, but the test and set (TS) of LNIOFLG2 indicated that this I/O was already queued.
5	During I/O processing, an LRA (load real address) instruction returned a non-zero return code indicating that the storage was not page-fixed.
6	An attempt was made to enqueue an I/O request on the free queue, but the test and set (TS) of LNIOFLG2 indicated that this I/O was already queued.
7	An attempt was made to enqueue an I/O request on an output queue, but the test and set (TS) of LNIOFLG2 indicated that this I/O was already queued.
8	An attempt was made to enqueue an I/O request on a free queue, but the test and set (TS) of LNIOFLG2 indicated that this I/O was already queued.
9	LIOREAD was called when another read was already active.
10	After successfully dynamically allocating an I/O device, the TIOT could not be located for it.
11	During driver initialization, an LRA (load real address) instruction returned a non-zero return code indicating that the storage was not page-fixed.
12	Termination was requested for a specific LNI but the LCSDEV resource element could not be located.
13	Termination was requested for a specific LNI but the LNI anchor block was not found in the table of active LNIs.
14	A loop was detected in the LNIO output chain.

Action: Collect all job output, including dumps, and contact Customer Support.

### ABEND D06 (U3334)

Reason: A DLBL I/O error is encountered while attempting to load a module.

Action: Examine the module to determine possible cause of problem or recreate module and restart Unicenter TCPaccess.

**ABEND E04 (U3588)**

Reason: The parameter list passed to PCOREG or PCOREF is invalid. Probable causes include bad storage length or storage address or an invalid PCORE entry code.

Action: Save all output from the job. Check the WTO and job logs for any related messages.

If the user is using:

- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
- The default applications, contact Customer Support

**ABEND E06 (U3590)**

Reason: While attempting to load a module, the module use count and chain of LD#M blocks do not agree.

Action: Save all output from the job. Contact Customer Support.

**ABEND E07 (U3591)**

Reason: A PTASK calls the PACCESS P-service with an invalid PENTRY code. An ABEND of x'E07' occurs. Unicenter TCPaccess terminates from this ABEND.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Check the parameter lists passed to the P-service. If the user is using the default applications, contact Customer Support.

**ABEND E0D (U3597)**

Reason: The PRSMGR P-service is called with an invalid entry code in register R0. The entry code is larger than the maximum allowable entry code. Register R3 in the dump should contain the entry code passed in R0 to PRSMGR

Action: Save all output from the job. Check the WTO and job logs for related messages.

If you are:

- Writing applications using P-services, check the logic of your PRSMGR calls and parameter lists
- Using the default applications, contact Customer Support

### ABEND E10 (U3600)

- Reason: The entry code parameter passed to PEXSRV is invalid. This ABEND is a result of a PEXCH P-service call. The PEXCH P-service call is valid only in an MVT environment. Application developers should not be using the PEXCH P-service. This ABEND should never occur.
- Action: Save all output from the job. Check the WTO and job logs for any related messages. Contact Customer Support.

### ABEND E11 (U3601)

- Reason: ACDYNAL is called with an invalid entry code parameter.
- Action: Save all output from the job. Check the WTO and job logs for any related messages.
- If the user is using:
- P-services to create an application, check the program logic of the failing PTASK and the parameter lists passed to the P-service
  - The default applications, contact Customer Support

### ABEND F06 (U3846)

- Reason: While attempting to load a module, an error occurs causing entry into ACLOAD debug exit.
- Action: Save all output from the job. Check the WTO and job logs for any related messages. Contact Customer Support.

### ABEND F10 (U3856)

- Reason: A call to IQCELLG for cell storage to be used as a work area fails. This ABEND is a result of a PEXCH P-service call. The PEXCH P-service call is valid only in an MVT environment. Application developers should not be using the PEXCH P-service. This ABEND should never occur.
- Action: Save all output from the job. Check the WTO and job logs for any related messages. Contact Customer Support.

**ABEND F11 (U3857)**

Reason: A call to ACDYNAL is made to perform a close but the close fails.

Action: Save all output from the job. Check the WTO and job logs for any related messages. Contact Customer Support.

## FTP Return Codes

Programs FTP and FTP2 can terminate and set these different return codes:

Code	Description
0	FTP successful
8	An error occurred in processing the FTP commands, a remote host disconnected, or a file transfer failed. Check the SYSPUT data output for the problem.
16	A file allocation error occurred. Check that the SYSGET and SYSPUT files are in the batch JCL. Also, check to make sure that the FIOS option was entered in the PARMs passed to FTP. When running FTP2 in batch, unless the NOA option is specified, a return code of 16 can be returned when there is no NETRC file.





# API Sample Application Messages

---

This chapter lists the messages generated by the FINGER, WHOIS, ACSHELLO, and TTCP sample API application programs.

It includes:

- [Socket WHOIS](#) – Messages generated by the socket version of the WHOIS sample program
- [C WHOIS](#) – Messages generated by the C library version of the WHOIS client program
- [C ACSHELLO](#) – Messages generated by the socket version of the ACSHELLO sample program
- [Socket ACSHELLO](#) – Messages generated by the socket version of the ACSHELLO sample program
- [FINGER](#) – Messages generated by the FINGER program
- [TTCP](#) – Messages generated by the TTCP sample program

## Socket WHOIS

This section lists the messages generated by the socket version of the WHOIS sample program. Most of these messages use the perror function call of the socket library to issue the message. Therefore these messages have an error-specific string appended to them. See the “Error Messages” chapter for descriptions of the perror error messages of the socket library.

If the WHOIS program executes successfully, it returns a value of zero. When the socket WHOIS program encounters an error, it immediately terminates execution with a return code of 16.

### **WHOIS NAME (name) {HOST(hostname)}**

Reason: Usage help message display if the WHOIS command is issued with invalid command line parameters. WHOIS must be called with a required parameter of NAME() and with the optional parameter of HOST() (name of host to request WHOIS service).

### **WHOIS: bind:**

Reason: The WHOIS program cannot bind a name to the socket used to communicate with the WHOIS server. See the “Error Messages” chapter for more information about the error message.

### **WHOIS: connect:**

Reason: The WHOIS program cannot establish a TCP connection with the WHOIS server. See the “Error Messages” chapter for more information about the error message.

### **WHOIS: read:**

Reason: The WHOIS program encountered an error when reading from the socket used to communicate with the WHOIS server. See the “Error Messages” chapter for more information about the error message.

### **WHOIS: socket:**

Reason: The WHOIS program cannot open a socket to use to connect to the WHOIS server. See the “Error Messages” chapter for more information about the error message.

### **WHOIS: whois/tcp: : unknown service**

Reason: The WHOIS client cannot determine the port number of the WHOIS server to connect to, due to a failure of the directory services to return a correct value. Check that the database contains an entry for the WHOIS service using TCP.

### **WHOIS: write:**

Reason: The WHOIS program encountered an error when writing to the socket used to communicate with the WHOIS server. See the “Error Messages” chapter for more information about the error message.

**WHOIS: *hostname*: host unknown**

Reason: The WHOIS program cannot find the name of the host specified by the command line parameter `HOST()` using the directory services feature.

## C WHOIS

This section lists messages generated by the C library version of the WHOIS client program. If the C WHOIS program completes successfully, it returns an error code of zero. If any errors are encountered, it returns a 16.

**WHOIS NAME (*name*) {HOST(*hostname*)} {SYSID *subsystem id*}**

Reason: This message is issued when a command parameter is formatted incorrectly. It is to provide the user the syntax of the command. WHOIS must be called with a required parameter of `NAME()` and with optional parameters of `HOST()` (name of host to request WHOIS service) and `SYSID` (subsystem ID of Unicenter TCPaccess).

**WHOIS: ACLOSE FAILED : general return code is *ret\_code* WHOIS: APCBERRC - *err\_code***

Reason: When terminating the user session with the API, the WHOIS program detected an error at the completion of the `ACLOSE`. *ret\_code* is the general return code of the `ACLOSE` macro; *err\_code* is the error code field of the APCB that was used to call `ACLOSE`.

**WHOIS: AOPEN FAILED : general return code is *ret\_code* WHOIS: APCBERRC - *err\_code***

Reason: These two lines of output indicate that the WHOIS program cannot open an application session with the API. *ret\_code* is the general return code of the `AOPEN` macro; *err\_code* is the error code field of the APCB used to call `AOPEN`.

**WHOIS: TBIND FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: The WHOIS program attempted to assign a name to its local endpoint using `TBIND`, but an error was encountered by the API. The `TBIND` macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

Reason: This error message may be followed by a message generated by the API. Read the API messages in *Prefixed Messages* for information about the message.

**WHOIS: TCLOSE FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: After completing the conversation with the WHOIS server, the client encountered an error when attempting to close the endpoint used for this session. The TCLOSE macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TCONFIRM FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: While attempting to confirm the connection sequence, the API returned an error to the WHOIS program. The TCONFIRM macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TCONNECT FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: When trying to initiate a connection with the WHOIS server, an error was encountered. The TCONNECT macro returned an error indicated by the general error code, displayed at the *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TCP/WHOIS service unknown****WHOIS: DIRSRV FAILED: general return code is *num* WHOIS: DPLRTNCD - *value***

Reason: These three lines of messages are printed if the WHOIS program cannot locate the WHOIS service port for TCP using the directory services. *num* is the number of the general return code returned by the DIRSRV macro expansion. *value* is the value of the return code field of the Directory Services Parameter List (DPL) used when DIRSRV is called.

**WHOIS: TOPEN FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: The WHOIS program attempted to open a transport endpoint through which to communicate to the WHOIS server. The TOPEN macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TRECVD FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: When receiving data from the WHOIS server, an error was detected. The TRECEIVE macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TRELACK FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: When attempting to acknowledge a release indication received from the WHOIS server, the client detected an error. The TRELACK macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TRELESE FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: When completing the orderly release between the client and the server, an error was detected by the WHOIS program. The TRELEASE macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: TSEND FAILED : general return code is *err\_code* WHOIS: TPLRTNCD - *ret\_code***

Reason: While transmitting data to the WHOIS server, an error was detected. The TSEND macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**WHOIS: *hostname* host unknown****WHOIS: DIRSRV FAILED: general return code is *num* WHOIS: DPLRTNCD - *value***

Reason: These three lines of messages are printed if the WHOIS program cannot locate the Network Information Center's name using the directory services. The hostname is replaced by the default host name of the NIC or by the name of the host specified by the user in the HOST command parameter. *num* is the number of the general return code returned by the DIRSRV macro expansion. *value* is the value of the return code field of the Directory Services Parameter List (DPL) used when DIRSRV is called.

## C ACSHELLO

This section lists those messages issued by the C library version of the ACSHELLO program. If the ACSHELLO program encounters any unrecoverable problem during its operation, it returns a code of 16. Because the program executes a loop servicing the clients requesting information, the program does not terminate successfully. You must either cancel the job or attention out the TSO command.

**ACSHELLO {PORT(TCP *port*)} {SYSID (*subsystem id*)}**

Reason: This message is issued when a command parameter is formatted incorrectly. It is to provide the user the syntax of the command. ACSHELLO may be called with no parameters or optionally with a well-known port number and/or a subsystem ID.

**ACSHELLO: AOPEN FAILED : general return code is *ret\_code*  
ACSHELLO: APCBERRC - *err\_code***

Reason: These two lines of output indicate that the ACSHELLO program cannot open an application session with the API. *ret\_code* is the general return code of the AOPEN macro, and *err\_code* is the error code field of the APCB that was used to call AOPEN.

**ACSHELLO: DIRSRV FAILED : general return code is *ret\_code*  
ACSHELLO: DPLRTNCD - *value***

Reason: The ACSHELLO server cannot look up its local host name and/or its internet address from NDS. *ret\_code* is the number of the general return code returned by the DIRSRV macro expansion. *value* is the value of the return code field of the Directory Services Parameter List (DPL) used when DIRSRV was called.

**ACSHELLO: TACCEPT FAILED : general return code is *err\_code*  
ACSHELLO: TPLRTNCD - *ret\_code***

Reason: On receiving a connection request from a client, the server attempted to accept the connection to a new endpoint. The TACCEPT macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TBIND FAILED : general return code is *err\_code*  
ACSHELLO: TPLRTNCD - *ret\_code***

Reason: The ACSHELLO program attempted to assign a name to its local endpoint using TBIND but an error was encountered by the API. The TBIND macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*. This error message may be followed by a message generated by the API. Read the API messages in *Prefixed Messages* for information about the message.

**ACSHELLO: TCLOSE FAILED : general return code is *err\_code*****ACSHELLO: TPLRTNCD - *ret\_code***

Reason: After providing service to a client, the ACSHELLO program detected an error on the TCLOSE used to destroy the local endpoint. The TCLOSE macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TLISTEN FAILED : general return code is *err\_code*****ACSHELLO: TPLRTNCD - *ret\_code***

Reason: The ACSHELLO program attempted to wait on the local endpoint receiving an incoming connection request from a client. The TLISTEN macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TOPEN FAILED : general return code is *err\_code*****ACSHELLO: TPLRTNCD - *ret\_code***

Reason: The ACSHELLO program attempted to open a transport endpoint through which to wait for incoming connection requests from clients. The TOPEN macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TRELACK FAILED : general return code is *err\_code*****ACSHELLO: TPLRTNCD - *ret\_code***

Reason: When acknowledging the completion of the orderly release from the client, the ACSHELLO server received an error on the release acknowledgment. The TRELACK macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TRELEASE FAILED : general return code is *err\_code*****ACSHELLO: TPLRTNCD - *ret\_code***

Reason: After issuing a TRELEASE request to the API to begin an orderly release of the TCP connection to the client, an error was returned by the API to the ACSHELLO program. The TRELEASE macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: TSEND FAILED : general return code is *err\_code***  
**ACSHELLO: TPLRTNCD - *ret\_code***

Reason: When trying to send data to a connected client, the ACSHELLO server was returned an error from the API. The TSEND macro returned an error indicated by the general error code, *err\_code*, and the Transport Parameter List (TPL) return code field, *ret\_code*.

**ACSHELLO: CPU ID is: *id\_num***

Reason: This message is transmitted to a connected client to indicate the CPU serial number and model number of the host on which ACSHELLO is running.

**ACSHELLO: Host's internet address is: *int\_addr***

Reason: This message is transmitted to a connected client to indicate the internet address used by this host to communicate.

**ACSHELLO: MVS SMF ID is: *smf\_id***

Reason: This message is transmitted to a connected client to indicate the SMF ID of the host on which ACSHELLO is running.

**ACSHELLO: MVS version level is: *MVS/vers***

Reason: This message is transmitted to a connected client to indicate the version level of MVS on which the ACSHELLO program is running.

**ACSHELLO: Running on host: *hostname***

Reason: This message is transmitted to a connected client to indicate the name of the host on which ACSHELLO is executing. *hostname* is the name of the host or the string No host name, if ACSHELLO encountered a problem getting its name from NDS.

**ACSHELLO: Server starting at: *date\_time***

Reason: This message is transmitted to a connected client to indicate that the ACSHELLO server has accepted its request for service and proceeds to do so. *date\_time* is the current date and time as reflected by the system on which ACSHELLO is running.



**ACSHELLO: Server stopping at: *date\_time***

Action: This message is transmitted to a connected client to indicate that the ACSHELLO has completed its service to the connected client and now terminates. *date\_time* is the current date and time as reflected by the system on which ACSHELLO is running.

## Socket ACSHELLO

This section lists the messages generated by the socket version of the ACSHELLO sample program. The ACSHELLO program runs in a loop, servicing clients as they request information of the server. Therefore it does not terminate with a successful return code. The user should cancel it or attention interrupt the TSO command. However if the ACSHELLO server detects an unrecoverable error, it terminates with a return code of 16.

Most of these messages use the perror function call of the socket library to issue the message and have an error-specific string appended to them. See the “Error Messages” chapter for more information.

**ACSHELLO {PORT(TCP port)}**

Reason: This message is issued when a command parameter is formatted incorrectly. It is to provide the user the syntax of the command. ACSHELLO may be called with no parameters or optionally with a well-known port number.

**ACSHELLO: accept**

Reason: The ACSHELLO program encountered an error when trying to accept an incoming connection from a client. See the “Error Messages” chapter for more information about the error message.

**ACSHELLO: bind**

Reason: The ACSHELLO program cannot bind a well-known name to the socket used to listen for requests from clients. See the “Error Messages” chapter for a more detailed explanation of the error message.

### **ACSHELLO: close**

Reason: The ACSHELLO program encountered an error when closing the connection to a client that just received servicing. See the “Error Messages” chapter for more information about the error message.

### **ACSHELLO: gethostbyname**

Reason: The ACSHELLO program encountered an error after issuing a request to NDS to retrieve its internet address.

### **ACSHELLO: gethostname**

Reason: The ACSHELLO program detected an error after issuing a request to NDS to retrieve the local host’s name. See the “Error Messages” chapter for more information about the error message.

### **ACSHELLO: listen**

Reason: The ACSHELLO program encountered an error when attempting to enable the socket for accepting connection requests. See the “Error Messages” chapter for more information about the error message.

### **ACSHELLO: select**

Reason: The ACSHELLO program detected an error when using the select socket call to perform synchronous I/O multiplexing. See the “Error Messages” chapter for more information about the error message.

### **ACSHELLO: socket**

Reason: The ACSHELLO program cannot open a socket to use to serve clients requesting service. See the “Error Messages” chapter for more information about the error message.

### **ACSHELLO: write**

Reason: The ACSHELLO program encountered an error when writing data to a connected client. See the “Error Messages” chapter for more information about the error message.

**ACSHELLO: CPU ID is...**

Reason: This message is transmitted to a connected client to indicate the CPU serial number and model number of the host on which ACSHELLO is running.

**ACSHELLO: Host's internet address is: *int\_add***

Reason: This message is transmitted to a connected client to indicate the internet address used by this host to communicate.

**ACSHELLO: MVS SMF ID is: *host\_id***

Reason: This message is transmitted to a connected client to indicate the SMF ID of the host on which ACSHELLO is running.

**ACSHELLO: MVS version level is: *MVS/level***

Reason: This message is transmitted to a connected client to indicate the version level of MVS on which the ACSHELLO program is running.

**ACSHELLO: Running on host: *hostname***

Reason: This message is transmitted to a connected client to indicate the name of the host on which ACSHELLO is executing. *hostname* is the name of the host or the string no host name, if ACSHELLO encountered a problem getting its name from NDS.

**ACSHELLO: Server starting at: *date\_time***

Reason: This message is transmitted to a connected client to indicate that the ACSHELLO server has accepted its request for service and proceeds to do so. *date\_time* is the current date and time as reflected by the system on which ACSHELLO is running.

**ACSHELLO: Server stopping at: *date\_time***

Reason: This message is transmitted to a connected client to indicate that the ACSHELLO has completed its service to the connected client and now terminates. The *date\_time* is replaced with the current date and time as reflected by the system on which ACSHELLO is running.

## FINGER

This section lists messages generated by the FINGER program. Most of these messages use the perror function call of the socket library to issue the message. Therefore these messages have an error-specific string appended to them. See the “Error Messages” chapter in this document more information.

If the FINGER program executes successfully, it returns a value of zero. When the FINGER program encounters an error, it immediately terminates execution with a return code of 16.

### FINGER NAME (USERID@HOST) {LONG}

Reason: Usage help message display if FINGER command is issued with invalid command line parameters. FINGER must be called with a required parameter of NAME() and with the optional parameter of LONG.

### FINGER: connect

Reason: The FINGER program could not establish a TCP connection with the FINGER server. Read the section on perror generated messages in the chapter “Error Messages” for a more detailed explanation of the error message.

### FINGER: socket

Reason: The FINGER program could not open a socket to use to connect to the FINGER server. Read the section on perror generated messages in the chapter “Error Messages” for a more detailed explanation of the error message.

### FINGER: TCP/FINGER: unknown service

Reason: The FINGER client could not determine the port number to connect to of the FINGER server, due to a failure of the directory services to return a correct value. The user should ensure that the database contains an entry for the FINGER service using TCP.

### FINGER: unknown host: *hostname*

Reason: The FINGER program could not find the name of the host specified by the command line parameter NAME() using the directory services feature.

**(hostname)**

Reason: The name of the host specified as part of the NAME() parameter has been located by the directory services.

## TTCP

This section lists messages generated by the TTCP sample program.

If the TTCP program executes successfully, it returns a value of zero. A return code of 100 (decimal) indicates that the MVS IKJPARS facility encountered a severe error in trying to parse the TTCP parameters. A return code of 52 (decimal) indicates a general TTCP program error and is accompanied by a message. A return code of 48 (decimal) indicates the failure of a TPL-based API operation and is accompanied by a message.

### **AOPEN FAILED; VERIFY THAT SSN PARM IS CORRECT AND TCP IS EXECUTING**

Reason: An AOPEN macro issued by TTCP failed. This generally means that the SSN parameter was incorrectly specified or that the specified Unicenter TCPaccess subsystem is not executing.

### **BAD DPL FUNCTION CODE (DPLFNCCD) FOUND AFTER DIRSRV ERROR**

Reason: After a DIRSRV macro issued by TTCP failed, the DPL function code (DPLFNCCD) was found to be invalid.

### **BAD TPL FUNCTION CODE (TPLFNCCD) FOUND AFTER T-FUNCTION ERROR**

Reason: After a TPL-based operation issued by TTCP failed, the TPL function code (TPLFNCCD) was found to be invalid.

### **CIB ADDRESS = 0 AFTER STOP (P) COMMAND**

Reason: When running TTCP under the TSO TMP in a batch job, the address of the CIB was found to be zero after an MVS STOP (P) command.

**CIB VERB NOT EQUAL TO CIBSTOP AFTER STOP (P) COMMAND**

Reason: When running TTCP under the TSO TMP in a batch job, the CIBVERB field in the CIB control block was found not to contain the expected CIBSTOP value after an MVS STOP (P) command.

**COMLIST ADDRESS = 0 AFTER EXTRACT COMM MACRO ISSUED**

Reason: When running TTCP under the TSO TMP in a batch job, the EXTRACT COMM macro resulted in the returned COMLIST address being zero.

**INVALID BUFLEN PARAMETER - MUST BE  $\leq$  TIB MAX (TIBLTRCV)**

Reason: The BUFLEN parameter entered by the user is invalid. BUFLEN must be less than or equal to the Transport Information Block (TIB) maximum size of TRECVR/TRECVFR data (TIBLTRCV) that is set via the MAXLTRECV parameter of the TCP or UDP statement.

**INVALID BUFLEN PARAMETER - MUST BE  $\leq$  TIB MAX (TIBLTSND)**

Reason: The BUFLEN parameter entered by the user is invalid. BUFLEN must be less than or equal to the Transport Information Block (TIB) maximum size of TSEND/TSENDTO data (TIBLTSND) that is set via the MAXLTSEND parameter of the TCP or UDP statement.

**INVALID BUFLEN PARAMETER - MUST BE  $>$  ZERO (0)**

Reason: The BUFLEN parameter entered by the user is invalid. When TTCP is in receive mode, BUFLEN must be greater than zero.

**INVALID BUFNUM PARAMETER - MUST BE  $\leq$  TIB MAX (TIBQRECV)**

Reason: The BUFNUM parameter entered by the user is invalid. BUFNUM must be less than or equal to the Transport Information Block (TIB) maximum size of the receive queue (TIBQRECV) that is set via the MAXQRECV parameter of the TCP or UDP statement.

**INVALID BUFNUM PARAMETER - MUST BE  $\leq$  TIB MAX (TIBQSEND)**

Reason: The BUFNUM parameter entered by the user is invalid. BUFNUM must be less than or equal to the Transport Information Block (TIB) maximum size of the send queue (TIBQSEND) that is set via the MAXQSEND parameter of the TCP or UDP statement.

**INVALID BUFNUM PARAMETER - MUST BE > ZERO (0)**

Reason: The BUFNUM parameter entered by the user is invalid. BUFNUM must be greater than zero.

**INVALID COMBINATION OF BUFNUM AND BUFLLEN PARAMETERS - MUST BE <= TIB MAX (TIBLRECV)**

Reason: While the individual BUFLLEN and BUFNUM parameters entered by the user are valid, the combination of the two is invalid. The product of BUFLLEN multiplied by BUFNUM must be less than or equal to the Transport Information Block (TIB) maximum size of the receive buffer (TIBLRECV) that is set via the MAXRECVBUF parameter of the TCP or UDP statement.

**INVALID COMBINATION of BUFNUM AND BUFLLEN PARAMETERS - MUST BE <= TIB MAX (TIBSEND)**

Reason: While the individual BUFLLEN and BUFNUM parameters entered by the user are valid, the combination of the two is invalid. The product of BUFLLEN multiplied by BUFNUM must be less than or equal to the Transport Information Block (TIB) maximum size of the send buffer (TIBSEND) that is set via the MAXSENDBUF parameter of the TCP or UDP statement.

**INVALID COUNT PARAMETER - MUST BE > ZERO (0)**

Reason: The count parameter entered by the user is invalid. When TTCP is in transmit mode, COUNT must be greater than zero.

**INVALID MESSAGE ID ID PASSED TO PUTLINE ROUTINE**

Reason: The TTCP subroutine that issues the TSO PUTLINE macro was passed an invalid message identifier.

**NO FREE RECEIVER SUBTASK FOUND WHERE EXPECTED**

Reason: In TTCP receive mode, a free receiver subtask control block was not found to allow passing of the endpoint to a receiver subtask.

**NULL (ZERO) ENDPOINT ID ID PASSED TO RECEIVER**

Reason: In TTCP receive mode, the receiver subtask was passed an invalid (null) endpoint ID.

**QEDIT CIBPTR=0 (DISALLOW MODIFY COMMANDS) FAILED**

Reason: When running TTCP under the TSO TMP in a batch job, the QEDIT CIBPTR=0 macro failed; the macro is issued to disallow the MVS MODIFY (F) command from being issued to the batch job.

**QEDIT FREE OF CIB FAILED**

Reason: When running TTCP under the TSO TMP in a batch job, the QEDIT FREE of the CIB failed; this CIB represents the MVS STOP (P) command.

**STAX ATTNEXIT FAILED**

Reason: When running TTCP under TSO (nonbatch) the STAX ATTNEXIT macro failed.

**TTCPR: TRANSFER SECONDS *secs* TRECVR's *trecvs*, TRECVR's/SEC *trecvsps* BYTES RECEIVED *bytes*, BYTES/SEC *bytesps***

Reason: Statistic message issued Transport Information Block (TIB) completion of a receive (sink) transfer.

Parameter	Description
<i>secs</i>	The number of seconds the transfer took.
<i>trecvs</i>	The number of TRECVR/TRECVRFR operations completed.
<i>trecvsps</i>	The number of TRECVR/TRECVRFR operations completed per second.
<i>bytes</i>	The number of data bytes received.
<i>bytesps</i>	The number of data bytes received per second.



**TTCPT: TRANSFER SECONDS secs TSEND's *tsends*, TSEND's/SEC *tsendsps* BYTES SENT *bytes*, BYTES/SEC *bytesps***

Reason:                      Statistic message issued Transport Information Block (TIB) completion of a transmit (source) transfer.

Parameter	Description
secs	The number of seconds the transfer took.
tsends	The number of TSEND/TSENDTO operations completed.
tsendsps	The number of TSEND/TSENDTO operations completed per second.
bytes	The number of data bytes received.
bytesps	The number of data bytes received per second.

**UNEXPECTED ENDPOINT STATE AFTER TOPEN OPTCD=OLD COMPLETED**

Reason:                      In TTCP receive mode, the receiver subtask found the passed endpoint to be in the wrong state.



# API/IFS Internal Trace Entries

This chapter contains internal API and IFS trace entries.

It includes these sections:

- [Flags and Return Codes](#) – Provides additional information about the SRE flags, TPL flags, and APICTFRE return codes
- [IFS Internal Trace Entries](#) – Describes IFS internal trace entries.

## Flags and Return Codes

The following section provides additional information on the SRE flags, TPL flags, and APICTFRE return codes.

### SRE Flags

The following is a list of all SRE flags:

Flag	Description
x'80'	Exit request.
x'40'	Special processing.
x'20'	Static SRE, not freed.

## TPL Flags

The following is a list of IPL flags:

Flag	Description
x'80'	Completed.
x'40'	Completed with error.
x'20'	External ECB.
x'10'	Completion exit routine.
x'08'	AMODE 31.
x'04'	Accepting connect.

## APICTFRE Return Codes

The following is a list of APICTFRE return codes:

Flag	Description
0	Normal completion.
4	Completion exit.
8	Flushing SREs.
12	Protocol event exit.
16	Failed, no XWAs.
20	Abortive close.

## IFS Internal Trace Entries

The following section describes IFS internal trace entries, listed alphabetically by type.

### ABND

The following is a list of internal trace entries produced by the ABEND capture routines:

Subtype	User Fields/Modules	Description
--	UDW1: PSW word 1 UDW2: PSW word 2 UDW3: Abend code Modules: IFSFDEBUG, IFSGDEBUG, IFSXESTA, IJTCCIB, T01AISFR	A program ABEND was captured.

### CALL

The following is a list of CALL-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
ADDR	UDW1: First four characters caller's name UDW2: Second four characters caller's name UDW3: First four characters callee's name UDW4: Second four characters callee's name UDW5: Call indicator flag Modules: IFSSCALL	Module called via entry point address.
ECM	UDW1: First four characters of module called. UDW2: Second four characters of module called. UDW3: Register 3 on entry the PSW key mask and the primary ASID UDW4: Register 4 on entry to the A(latent parameter list) used to locate the ICVT UDW5: Register 1 parameter value on entry Modules: IJTOPCNM, IFSCSSI, IJTCCIB	Module called via external cross memory PC (space switch).

Subtype	User Fields/Modules	Description
ECMR	UDW1: First four characters of module exiting UDW2: Second four characters of module exiting UDW3: Register 3 on entry to the PSW key mask and the primary ASID UDW4: Register 4 on entry to the A(latent parameter list) used to locate the ICVT UDW5: Not used Modules: IJTOPCNM, IFSCSSI, IJTCCIB	Module called via external cross memory PC is returning to the caller.
LCD	UDW1: First four characters caller's name UDW2: Second four characters caller's name UDW3: First four characters callee's name UDW4: Second four characters callee's name Modules: IFSSCALL	Module called via LOAD, CALL, and DELETE (non-memory resident module)
LCM	UDW1: First four characters caller's name UDW2: Second four characters caller's name UDW3: First four characters callee's name UDW4: Second four characters callee's name UDW5: Call indicator flag Modules: IFSSCALL	Module called via local cross memory PC (non- space switch).
LDER	UDW1: First four characters callee's name UDW2: Second four characters callee's name UDW3: Register 15 set by callee UDW4: Register 0 set by callee UDW5: Register 1 set by callee Modules: IFSSCALL	Module to be called was not found.
LDTB	UDW1: First four characters caller's name UDW2: Second four characters caller's name UDW3: First four characters callee's name UDW4: Second four characters callee's name UDW5: Call indicator fla Modules: IFSSCALL	Module called via loaded-module table entry.

Subtype	User Fields/Modules	Description
RET	UDW1: First four characters callee's name UDW2: Second four characters callee's name UDW3: Register 15 set by callee UDW4: Register 0 set by callee UDW5: Register 1 set by callee Modules: IFSSCALL	Module called is returning to caller.
SRB	UDW1: First four characters callee's name UDW2: Second four characters callee's name UDW3: Register 0 parameter on entry A(SRB) UDW4: Register 1 parameter on entry UDW5: Not used Modules: IFSVDAFY, IFSVIRPL, IFSVLGON, IFSVLTRM, IFSVNSXT, IFSVRELRL, IFSVSCIP, IFSVTPND	Module called via MVS SRB dispatch.
SRBR	UDW1: First four characters callee's name UDW2: Second four characters callee's name IDW3: Register 0 parameter on entry A(SRB) IDW4: Register 1 parameter on entry IDW5: Not used Modules: IFSVDAFY, IFSVIRPL, IFSVLGON, IFSVLTRM, IFSVNSXT, IFSVRELRL, IFSVSCIP, IFSVTPND	Module called via MVS SRB dispatch returning.

## CAP

The following is a list of internal trace entries produced by the trace capture routines:

Subtype	User Fields/Modules	Description
CBFE	UDW1: Caller's address space ID UDW2: Reason code UDW3: Return code Module: T03CTCAP	Error flushing trace buffer.
CDAT	UDW1: Caller's address space ID UDW2: Instance ID UDW3: Available buffer space UDW4: CTE address Module: T03CTCAP	Creating trace data headers.
CFLH	UDW1: Caller's address space ID UDW2: Instance ID UDW3: Buffer address UDW4: TBUF-full exit address Module: T03CTCAP	Flushing last full trace buffer.
CFLT	UDW1: Caller's address space ID UDW2: Instance ID UDW3: Record count UDW4: Maximum record count Module: T03CTCAP	Trace data passed filtering criteria.
CRQT	UDW1: Caller's address space ID UDW2: Group ID UDW3: Group type UDW4: Total record size Module: T03CTCAP	Initial trace point request.



**DISP**

The following is a list of DISP-type IFS internal trace entries:

<b>Subtype</b>	<b>User Fields/Modules</b>	<b>Description</b>
DRVR	UDW1: Address of FRR Stack UDW2: Address of first module work UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFSSTDRV	ISRB Dispatching – dispatch task driver.
INIT	UDW1: Address of FRR Stack UDW2: Address of first module work area (MWA) UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFFSSTDRV	ISRB Dispatching – initial dispatch of an ISRB.
RESD	UDW1: Address of FRR Stack UDW2: Address of first module work area (MWA) UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFSSTDRV	ISRB Dispatching – driver task posted.
RESM	UDW1: Address of FRR Stack UDW2: Address of first module work area (MWA) UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target	ISRB Dispatching – Resume dispatch of a previously suspended ISRB.

Subtype	User Fields/Modules	Description
	routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFSSTDV	
SUSP	UDW1: Address of FRR Stack UDW2: Address of first module work area (MWA) UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFSSTDV	ISRB Dispatching – Suspend dispatch of an ISRB.
TERM	UDW1: Address of FRR Stack UDW2: Address of first module work area (MWA) UDW3: Two-byte queue priority, one-byte lock mask UDW4: Name or address of target routine UDW5: Characters five-eight of routine (UDW4-addr) Modules: IFSSTDV	ISRB Dispatching – Terminate dispatch of an ISRB.

## DOWN

The following is a list of internal trace entries produced by the trace instance termination routines:

Subtype	User Fields/Modules	Description
DBFE	UDW1: Address space ID UDW2: Reason code UDW3: Return code Module: T03CTTDN	Error flushing trace buffer.
DFLH	UDW1: Address space ID UDW2: Instance ID UDW3: Buffer address Module: T03CTTDN	Flushing last full trace buffer.
DRQT	UDW1: Address space ID UDW2: Instance ID Module: T03CTTDN	Initial trace instance stop request.

## DRIV

The following is a list of internal trace entries produced by the drivers:

Subtype	User Fields/Modules	Description
CETI	UDW1: Input control block header UDW2: Inbound-outbound port ACKs UDW3: Output control block header UDW4: Inbound-outbound port indexes Module: T01SLPCS	Starting CETI input/output cycle.
CETR	UDW1: Input control block header UDW2: Inbound / outbound port ACKs UDW3: Output control block header UDW4: Inbound / outbound port indexes Module: T01SLPCS	Ending CETI input/output cycle.
DIEX	UDW1: Unit name UDW2: LNSI address UDW3: Return code UDW4: SRB address Module: T01SLDIE	Exiting DIE routine.

Subtype	User Fields/Modules	Description
FEWB	UDW1: First empty write-LNIO address UDW2: Last empty write-LNIO address UDW3: Current empty write-LNIO address UDW4: First active write-LNIO address UDW5: Last active write-LNIO address Module: T01SLDIE	Received CLAW write-port interrupt.
FFRB	UDW1: First filled read-LNIO address UDW2: Last filled read-LNIO address UDW3: First active read-LNIO address UDW4: Last active read-LNIO address Module: T01SLDIE	Received CLAW read-port interrupt.
IO	UDW1: Unit name UDW2: LNSI address UDW3: CSW word 1 UDW4: CSW word 2 Module: T01SLDIE	Entering DIE routine.
GTWR	UDW1: Unit name UDW2: Return code UDW3: Buffer address UDW4: Length of buffer Module: T01SLGTW	Getting I/O buffer for write.
PTWR	UDW1: Unit name UDW2: Filled buffer count UDW3: Old output user count UDW4: New output user count Module: T01SLPTW	Queuing I/O buffer for write.
SIO	UDW1: Unit name UDW2: LNUI address UDW3: Real address UDW4: Virtual address Module: T01SLSIO	Issuing STARTIO.
SIOD	UDW1: Unit name UDW2: LNUI address UDW3: Real address UDW4: Virtual address Module: T01SLSIO	Building LNSI for STARTIO.

Subtype	User Fields/Modules	Description
UI	UDW1: Unit name UDW2: CSW word 1 UDW3: CSW word 2 Module: T01SLUIH	Received unsolicited interrupt (unit check, device end, unsolicited attention).

## EXIT

The following is a list of internal trace entries produced by the exit handlers:

Subtype	User Fields/Modules	Description
ENTR	UDW1: Exit name part 1 UDW2: Exit name part 2 UDW3: Context field UDW4: Caller parameter 1 UDW5: Caller parameter 2 Module: IFSSEXIT	Calling user exit routine.
EXIT	UDW1: Exit name part 1 UDW2: Exit name part 2 UDW3: Context field UDW4: Return code UDW5: Exit return code Module: IFSSEXIT	Returned from user exit routine.

## IUCV

The following is a list of IFS internal trace entries for IUCV.

Type	User Fields	Description	Module
CLBK	UDW1: Pathid UDW2: Function (text) UDW3: A(SEPM) UDW4: A(SAW)	IUCV callback issued.	T01EUxx T01AMIUC T011SRV T011SCA T011SGS T011STS
CONT	UDW1: Pathid UDW2: Function (text) UDW3: A(SEPM) UDW4: A(SAW)	Function resumed.	T011SPC
DEQU	UDW1: A(IASB) UDW2: A(PTPC) UDW3: A(IUMS)	IUMS dequeued from IASB or PTPC.	T02SSCHD T02CINTL
DIRC	UDW1: A(IUMS) UDW2: PC number UDW3: IPARML+0 UDW4: IPARML+4	Calling direct call external interrupt routine (Part 1).	T02SSCHD
DIR2	R15: IPARML+8 R00: IPARML+C R01: IPARML+10 UDW1: IPARML+14 UDW2: IPARML+18 UDW3: IPARML+1C UDW4: IPARML+20	Calling direct call external interrupt routine (Part 2).	T02SSCHD
DIRX	UDW1: A(IUMS) UDW2: PC number	Return for call to direct call external interrupt routine.	T02SSCHD
FAIL	UDW1: Pathid UDW2: Function (text) UDW3: Return code UDW4: IPRCODE	IUCV function call failed.	T011Sxx T01AMIUC

Type	User Fields	Description	Module
FARG	R15: TRGCLS, socket number R00: Argument data+0 R01: Argument data+4 UDW1: Argument data+8 UDW2: Argument data+C UDW3: Argument data+10 UDW4: Argument data+14	IUCV socket function call data (contents of argument data depends on function).  The TRGCLS information contained in R15 is described in the TRGCLS table, following.	T011Sxx
FREP	R15: TRGCLS, socket number R00: Argument data+0 R01: Argument data+4 UDW1: Argument data+8 UDW2: Argument data+C UDW3: Argument data+10 UDW4: Argument data+14	IUCV socket function reply data (contents of argument data depends on function).	T011Sxx
FRET	UDW1: A(IPARML) UDW2: Return_value UDW3: IPARML+0 UDW4: IPARML+4	IUCV function complete (part 1).	T02CIUCV
FRE2	R15: IPARML+8 R00: IPARML+C R01: IPARML+10 UDW1: IPARML+14 UDW2: IPARML+18 UDW3: IPARML+1C	IUCV function complete (part 2).	T02CIUCV

Type	User Fields	Description	Module
	UDW4: IPARML+20		
FUNC	UDW1: A(IPARML) UDW2: Function (text) UDW3: IPARML+0 UDW4: IPARML+4	IUCV function being processed (part 1).	T02CIUCV
FUN2	R15: IPARML+8 R00: IPARML+C R01: IPARML+10 UDW1: IPARML+14 UDW2: IPARML+18 UDW3: IPARML+1C UDW4: IPARML+20	IUCV function being processed (part 2).	T02CIUCV
IENT	UDW1: Pathid UDW2: Function (text) UDW3: A(SEPM)	Socket function being processed by OMVS	T011SPC
IEXT	UDW1: Pathid UDW2: Function (text) UDW3: Return_value UDW4: Return_code	Socket function has completed processing.	T011SPC
INTL	UDW1: A(IRBI) UDW2: A(ASCB) UDW3: A(TCB)	IUCV internal function call.	T02CINTL
INTX	UDW1: A(IRBI) UDW2: Return_value UDW3: IPARML+0 UDW4: IPARML+4	IUCV internal function call return (part 1).	T02CINTL



Type	User Fields	Description	Module
INT2	R15: IPARML+8 R00: IPARML+C R01: IPARML+10 UDW1: IPARML+14 UDW2: IPARML+18 UDW3: IPARML+1C UDW4: IPARML+20	IUCV internal function call return (part 2).	T02CINTL
PCAB	UDW1: PSW word 1 UDW2: PSW word 2 UDW3: PSW word 3 UDW4: PSW word 4	Abend occurred during IUCV processing.	T02GIUCV
QUED	UDW1: A(IASB) UDW2: A(PTPC) UDW3: A(IUMS) UDW4: A(EORA)	IUMS queued to IASB or PTPC.	T02SSCHD
SIRB	UDW1: A(IASB) UDW2: A(SRBI) or 0 UDW3: A(IRBI)	External interrupt scheduled.	T02SSCHD
WAIT	UDW1: Pathid UDW2: Function (text) UDW3: A(SEPM) UDW4: A(SAW)	Socket function is awaiting callback.	T011Sxx

**Note:** The maximum internal trace table size is now 1024 pages.

**TRGCLS Table**

TRGCLS function call values (for use in interpreting Internal Traces):

<b>Value</b>	<b>Description</b>
0	INITAPI message
1	Socket Function: accept()
2	Socket Function: bind()
3	Socket Function: close()
4	Socket Function: connect()
5	Socket Function: fcntl()
7	Socket Function: gethostid()
8	Socket Function: gethostname()
9	Socket Function: getpeername()
10	Socket Function: getsockname()
11	Socket Function: getsockopt()
12	Socket Function: ioctl()
13	Socket Function: listen()
14	Socket Function: read(), readv()
16	Socket Function: recv(), recvfrom()
19	Socket Function: select()
20	Socket Function: send()
22	Socket Function: sendto()
23	Socket Function: setsockopt()
24	Socket Function: shutdown()
25	Socket Function: socket()
26	Socket Function: write()
29	LastErrNo Function
30	Socket Function: getclientid()
31	Socket Function: givesocket()
32	Socket Function: takesocket()
35	Socket Function: getibmssockopt()

Value	Description
36	Socket Function: setibmssockopt()
42	Cancel Function

## LOCK

The following is a list of LOCK-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
CCB	UDW1: One byte each Action, Mode, Lock Type Action Obtain(0), Release(4), Test(8) Mode Conditional(0), Unconditional(4) Type Local(0), CML(4), Control Block(8) 4th byte Reserved. UDW2-5: Not used. Modules: IFSSLOCK	Lock Service – Conditionally obtain a control block lock.
CCML	UDW1: One byte each Action, Mode, Lock Type Action Obtain(0), Release(4), Test(8) Mode Conditional(0), Unconditional(4) Type Local(0), CML(4), Control Block(8) 4th byte Reserved. UDW2-5: Not used. Modules: IFSSLOCK	Lock Service – Conditionally obtain the cross memory local lock.
CLOC	UDW1: One byte each Action, Mode, Lock Type Action Obtain(0), Release(4), Test(8) Mode Conditional(0), Unconditional(4) Type Local(0), CML(4), Control Block(8) 4th byte Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Conditionally obtain the local lock.

Subtype	User Fields/Modules	Description
OCB	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Unconditionally obtain a control block lock.
OCML	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Unconditionally obtain the cross memory local lock.
OLOC	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Unconditionally obtain the local lock.
RCB	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Release a control block lock.
RCML	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Release the cross memory local lock.

Subtype	User Fields/Modules	Description
RLOC	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Release the local lock.
TCB	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Test status of a control block lock.
TCML	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Test status of cross memory local lock.
TLOC	UDW1: One byte each    Action, Mode, Lock Type Action   Obtain(0), Release(4), Test(8) Mode   Conditional(0), Unconditional(4) Type   Local(0), CML(4), Control Block(8) 4th byte   Reserved UDW2-5: Not used Modules: IFSSLOCK	Lock Service – Test status of local lock.

## MSG

The following is a list of MSG-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
	UDW1: Length of WTO and MCS flags UDW2: Characters one to four of message number issued UDW3: Characters five and six of message number issued UDW4: Two half words of Descriptor and Route Code UDW5: WTO Multi-line information Modules: IFSSMSG	Operator message issued.

## OE

The following is a list of internal trace entries produced by OpenEdition:

Subtype	User Fields/Modules	Description
CONT	UDW1: SEPM address UDW2: Function code UDW3: SAW address UDW4: ASCB address Module: T010SPC	Continuing an asynchronous function
FENT	UDW1: SEPM address UDW2: Function code UDW3: INODE address UDW4: OE parm list address Module: T010SPC	Beginning specific function processing.
FEXT	UDW1: SEPM address UDW2: Function code UDW3: Return value UDW4: Reason code Module: T010SPC	Returning to the user.

Subtype	User Fields/Modules	Description
POST	UDW1: SEPM address UDW2: SAW address UDW3: Return value UDW4: Reason code  Modules: T01ESCF, T01ESCI, T01ESCM, T01ESDA, T01ESDI, T01ESRL, T01ESTP, T01ESWC	Posting completion of an asynchronous function.
SCHD	UDW1: SEPM address UDW2: SAW address UDW3: ASCB address UDW4: TCB address  Modules: T01ESCF, T01ESCI, T01ESCM, T01ESDA, T01ESDI, T01ESRL, T01ESTP, T01ESWC, T010SCA, T010SFR, T010SRD, T010SRM, T010SRV, T010SSD, T010SSM, T010STO, T010SVR, T010SVW, T010SWR	Scheduling an asynchronous action.
SELP	UDW1: SEPM address UDW2: SAW address UDW3: Return value  Modules: T01ESCF, T01ESCI, T01ESCM, T01ESDA, T01ESDI, T01ESRL, T01ESTP, T01ESWC, T01SSELP	Posting completion of an asynchronous select.
UPDA	UDW1: SEPM address UDW2: SAW address UDW3: ASCB address UDW4: TCB address  Modules: T010SAC, T010SCN, T010SFR, T010SRD, T010SRM, T010SRV, T010SSD, T010SSM, T010STO, T010SVR, T010SVW, T010SWR	Updating an asynchronous request with the PFS's request token.
WAIT	UDW1: SEPM address UDW2: SAW address UDW3: ASCB address UDW4: TCB address  Modules: T010SAC, T010SCL, T010SCN, T010SFR, T010SRD, T010SRM, T010SRV, T010SSD, T010SSM, T010STO, T010SVR, T010SVW, T010SWR	Waiting for completion of an asynchronous function.

Subtype	User Fields/Modules	Description
WAKE	UDW1: SEPM address UDW2: SAW address UDW3: Return code  Modules: T010SAC, T010SCL, T010SCN, T010SFR, T010SRD, T010SRM, T010SRV, T010SSD, T010SSM, T010STO, T010SVR, T010SVW, T010SWR	Resuming execution of an asynchronous function.

## PAGE

The following is a list of PAGE-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
FIXF	UDW1: Ending A(Object) being fixed. UDW2: Beginning A(Object) being fixed. UDW3-UDW4: Eight-character name of object type being fixed Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'FIXF' Modules: IFSSPAGE	Page Service – Fix a page(s) fast.
FIXL	UDW1: Ending A(Object) being fixed. UDW2: Beginning A(Object) being fixed. UDW3-UDW4: Eight-character name of object type being fixed Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'FIXL' Modules: IFSSPAGE	Page Service – Fix a page(s) fast.
FIXS	UDW1: Ending A(Object) being fixed. UDW2: Beginning A(Object) being fixed. UDW3-UDW4: Eight-character name of object type being fixed Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'FIXS' Modules: IFSSPAGE	Page Service – Fix a page(s) short.



Subtype	User Fields/Modules	Description
FREA	UDW1: Ending A(Object) being freed. UDW2: Beginning A(Object) being freed. UDW3-UDW4: Eight-character name of object type being freed Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'FREA' Modules: IFSSPAGE	Page Service – Free a page(s) fast.
FREE	UDW1: Ending A(Object) being freed. UDW2: Beginning A(Object) being freed. UDW3-UDW4: Eight-character name of object type being freed Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'FREE' Modules: IFSSPAGE	
LOAD	UDW1: Ending A(Object) being loaded. UDW2: Beginning A(Object) being loaded. UDW3-UDW4: Eight-character name of object type being loaded Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'LOAD' Modules: IFSSPAGE	Page Service – Force a page(s) load.
OUT	UDW1: Ending A(Object) being paged out. UDW2: Beginning A(Object) being paged out. UDW3-UDW4: Eight-character name of object type being paged out Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string C'OUT ' Modules: IFSSPAGE	Page Service – Force a page(s) out.

Subtype	User Fields/Modules	Description
RLSE	UDW1: Ending A(Object) being released. UDW2: Beginning A(Object) being released. UDW3-UDW4: Eight-character name of object type being released Module name, MWA, BUFFER, MODULE (calling module), AREA of control block ID. UDW5: Character string 'RLSE' Modules: IFSSPAGE	Page Service – Release a page(s).

## PMGR

The following is a list of PMGR-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
FPOL	UDW1: Pool header name UDW2: Number of free elements in the pool UDW3: A(first element being freed) UDW4-UDW5: Unused Modules: IFSPFPOL	Pool Management – Free a chain of pool elements.
GPOL	UDW1: Pool header name UDW2: Number of free elements in the pool UDW3: A(first element being freed) UDW4-UDW5: Unused Modules: IFSPGPOL	Pool Management – Get a chain of pool elements.

## POST

The following is a list of POST-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
	UDW1-UDW5: Not used Modules: IFSSTDV	Task driver posted.

## SAPI

The following is a list of internal trace entries produced by the SAPI interface:

Subtype	User Fields/Modules	Description
CONF	UDW1: SEPM address UDW2: Local host IP address UDW3: Remote host IP address UDW4: Local / remote port numbers Module: T01ASCNF	An outbound connection request was confirmed.
CONN	UDW1: SEPM address UDW2: Local host IP address UDW3: Remote host IP address UDW4: Local / remote port numbers Module: T01XCONN	An outbound connection request has been issued.
CON1	UDW1: SEPM address UDW2: Child SEPM address UDW3: Local host IP address UDW4: Remote host IP address UDW5: Local / remote port numbers Module: T01ASCON	An incoming request for a connection was received.
CON2	UDW1: SEPM address UDW2: Child SEPM address UDW3: Local host IP address UDW4: Remote host IP address UDW5: Local / remote port numbers Module: T01ASCON	An incoming request for a connection was completed.
CREA	UDW1: SEPM address UDW2: Access method / protocol UDW3: Address space ID Module: T01XCREA	An endpoint was created.

Subtype	User Fields/Modules	Description
DISC	UDW1: SEPM address UDW2: Reason code Module: T01ASDIS	A disconnect indication was received.
FREE	UDW1: SEPM address UDW2: Access method / protocol UDW3: Address space ID UDW4: Flags / PC count Modules: T01XFREE	An endpoint is being freed.
MBUF	UDW1: MBUF count UDW2: MBUF address UDW3: Byte count Modules: T01SMBUF	A request for an MBUF was made.
RECV	UDW1: SEPM address UDW2: MBUF address UDW3: MBUFRLen Module: T01ASRCV	Data was received and is being queued to an endpoint.
RELS	UDW1: SEPM address Module: T01ASREL	A release indication was received.
WRDN	UDW1: SEPM address UDW2: MBUF address UDW3: SAW address UDW4: Data length UDW5: Reason code Module: T01ASWDN	A write request was completed.

## SCHD

The following is a list of SCHD-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
COMP	UDW1: Target task group ID, three characters UDW2: Name or address of target routing UDW3: Characters five to eight of routing (UDW4-addr) UDW4: A(ISRB) being scheduled UDW5: Not used Modules: IFSSSCHD	ISRB Scheduling – prepare an extended ECB.
IMME	UDW1: Target task group ID, three characters UDW2: Name or address of target routing UDW3: Characters five to eight of routing (UDW4-addr) UDW4: A(ISRB) being scheduled UDW5: Not used Modules: IFSSSCHD	ISRB Scheduling – Immediately queue the ISRB to the target ITGB.
PREP	UDW1: Target task group ID, three characters UDW2: Name or address of target routing UDW3: Characters five to eight of routing (UDW4-addr) UDW4: A(ISRB) being scheduled UDW5: Not used Modules: IFSSSCHD	ISRB Scheduling – Prepare the ISRB ECB to be posted by an external process. If already posted, treat as an IMME type event.
XECB	UDW1: Target task group ID, three characters UDW2: Name or address of target routing UDW3: Characters five to eight of routing (UDW4-addr) UDW4: A(ISRB) being scheduled UDW5: Not used Modules: IFSXPOST	ISRB Scheduling – an extended ECB was posted.

## SSOB

The following is a list of SSOB-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
CMD	UDW1: Length of SSOB UDW2: A(SSIB) UDW3: Return code from subsystem UDW4: A(Function dependent area SSOB) UDW5: Not used Modules: IFSCSSI	Operator command received via subsystem interface.

## STACK

The following is a list of internal trace entries produced by the stack:

Subtype	User Fields/Modules	Description
DIMB	UDW1: SEPM address UDW2: MBUF address UDW3: Reason code Modules: T01SIMUX, T01SISND, T01SRIN, T01SRSND, T01STREA, T01SUIN, T01SUSND	An MBUF was discarded.
ENMB	UDW1: SEPM address UDW2: MBUF address UDW3: Address space ID Module: T01SUIN	An MBUF was queued to an endpoint.
FOMB	UDW1: Packet ID UDW2: MBUF address UDW3: Packet Length UDW4: Remote host IP address Module: T01SIGWY	A datagram was forwarded to another host.
ICIN	UDW1: ICMP type UDW2: MBUF address UDW3: Message length UDW4: ICMP subtype UDW5: Remote host IP address Module: T01SICMP	An ICMP datagram was received.
ICOU	UDW1: ICMP type UDW2: MBUF address	An ICMP datagram was sent.

Subtype	User Fields/Modules	Description
	UDW3: Message length UDW4: ICMP subtype UDW5: Remote host IP address Module: T01SICMS	sent.
IPFO	UDW1: Packet ID UDW2: MBUF address UDW3: Packet length UDW4: Flags and fragment offset Module: T01SISND	An IP fragment was sent.
IPIN	UDW1: Packet ID UDW2: MBUF address UDW3: Packet length UDW4: Sequence number Module: T01SIMUX	An IP datagram was received.
IPOU	UDW1: SEPM address UDW2: MBUF address UDW3: MEDA address UDW4: LNIC address UDW5: First-hop IP address Module: T01SISND	An IP datagram was sent.
IPRA	UDW1: Packet ID UDW2: MBUF address UDW3: Packet length UDW4: Offset to first data byte Module: T01SITIM	An IP datagram fragment was received and is being reassembled.
RWIN	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Protocol UDW5: Remote host IP address Module: T01SIMUX	A RAW datagram was received.
RWOU	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Protocol UDW5: Remote host IP address Modules: T01SRSND	A RAW datagram was sent.
TCCL	UDW1: SEPM address UDW2: <zero> UDW3: Child SEPM address	A TCP endpoint is being closed.

Subtype	User Fields/Modules	Description
	UDW4: ATCB address Module: T01STCLO	
TCIN	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Sequence number UDW5: TCP header flags R15: Acknowledgement number R00: Local / remote port numbers R01: Checksum / urgent pointer Module: T01STIN	A TCP datagram was received.
TCOU	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Sequence number UDW5: TCP header flags R15: Acknowledgement number R00: Local / remote port numbers R01: Checksum / urgent pointer Modules: T01STRST, T01STSND, T01STTMK	A TCP datagram was sent.
TCTI	UDW1: SEPM address UDW2: Shift count UDW3: Function code UDW4: Connection state Modules: T01STTMD, T01STTMK, T01STTML, T01STTMP, T01STTMR, T01STTMT	A TCP timer expired.
UDIN	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Local port UDW5: Remote port Module: T01SIMUX	A UDP datagram was received.
UDOU	UDW1: SEPM address UDW2: MBUF address UDW3: Data length UDW4: Local port UDW5: Remote port Module: T01SUSND	A UDP datagram was sent.



## TIME

The following is a list of internal trace entries produced by the timer routines:

Subtype	User Fields/Modules	Description
CBEX	UDW1: ITMC address UDW2: ITIM address UDW3: Exit address UDW4: Callback routine parm address Module: IFSRTIME	Timer deallocation called a callback exit.
CECB	UDW1: ITMC address UDW2: ITIM address UDW3: ECB address Module: IFSRTIME	Timer deallocation posted a callback ECB.
CRET	UDW1: ITMC address UDW2: ITIM address UDW3: Exit address UDW4: Callback routine parm address Module: IFSRTIME	A timer callback exit returned control to timer services.
TECB	UDW1: ITMC address UDW2: ITIM address UDW3: ECB address UDW4: Slot number Module: IFSRTIME	A timer expired and an ECB has been posted.
TRET	UDW1: ITMC address UDW2: ITIM address UDW3: Exit address UDW4: Callback routine parm address UDW5: Slot number Module: IFSRTIME	A timer exit returned control to timer services.
TXIT	UDW1: ITMC address UDW2: ITIM address UDW3: Exit address UDW4: Callback routine parm address UDW5: Slot number Module: IFSRTIME	A timer expired and an exit has been called.

**TLI**

The following table lists internal trace entries produced by the Transport Layer Interface:

<b>Subtype</b>	<b>User Fields/Modules</b>	<b>Description</b>
CLBK	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN  Modules: T01ETCF, T01ETCI, T01ETCM, T01ETDA, T01ETDI, T01ETRL, T01ETTP, T01ETWC, T012TSND	Processing a callback request.
COMP	UDW1: Function code UDW2: Return value UDW3: Reason code UDW4: TPL address UDW5: HASN / SASN  Module: T012SPC	Function is complete.
CONT	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN  Module: T012SPC	Continuing an asynchronous function.
ECBP	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: ECB address UDW5: HASN / SASN  Modules: T012VPRO, T012VTPL	Posting an ECB.
ENTR	UDW1: Function code UDW2: Parm UDW3: HASN / SASN  Module: T012SPC	Beginning general function processing.

Subtype	User Fields/Modules	Description
EOT	UDW1: Type UDW2: ISRB address UDW3: ATCB address UDW4: HASN / SASN Modules: T01CSSIX, T01SSSIX, T012RSUM	MVS task is ending.
EVNT	UDW1: Event code UDW2: SAW address UDW3: SEPM address UDW4: ATCB address UDW5: HASN / SASN Module: T012SPC	Processing protocol event.
EXIT	UDW1: Function code UDW2: Return value UDW3: Reason code UDW4: Parm UDW5: HASN / SASN Module: T012SPC	Returning to the user.
FUNC	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN Modules: T012SPC, T012TACC, T012TADR, T012TBIN, T012TCHK, T012TCLR, T012TCLS, T012TCNF, T012TCON, T012TDIS, T012TERR, T012TINF, T012TLIS, T012TOPN, T012TOPT, T012TRCT, T012TRCV, T012TRER, T012TRFR, T012TRJT, T012TRLK, T012TRLS, T012TSND, T012TSTA, T012TSTO, T012TUNB, T012TUSR	Beginning specific function processing.
IRBX	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: Exit address UDW5: HASN / SASN Modules: T012VPRO, T012VTPL	Beginning IRB exit processing.

Subtype	User Fields/Modules	Description
QEVT	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: HASN / SASN  Modules: T01ETCF, T01ETCI, T01ETDA, T01ETDI, T01ETRL, T01ETTP, T01ETWC	Chaining a SAW on the event queue.
RSUM	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN  Module: T012RSUM	Resuming a request block.
SCHD	UDW1: <zero> UDW2: SAW address UDW3: <zero> UDW4: ATCB address UDW5: HASN / SASN  Module: T012SSRB	Scheduling an SRB.
SRBX	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: Exit address UDW5: HASN / SASN  Modules: T012VPRO, T012VTPL	Beginning SRB exit processing
SUSP	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN  Module: T012SUSP	Suspending a request block.
WAIT	UDW1: Function code UDW2: SAW address UDW3: SEPM address UDW4: TPL address UDW5: HASN / SASN  Modules: T012TCLR, T012TCLS, T012TCNF, T012TDIS, T012TLIS, T012TOPN, T012TRCV, T012TRFR, T012TRLK, T012TRLS, T012TSND, T012TSTO	Requesting callback for an asynchronous function.

## TSO

The following is a list of ISO-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
PARS	UDW1: First four characters of command text UDW2: Second four characters of command text UDW3: A(Area) returned by IKJPARS UDW4: Length of area returned by IKJPARS UDW5: Not used Modules: IFSIPOOL, IFSO\$CON, IFSO\$ROU, IFSOABND, IFSOSNAP, IJTICNFG, IJTOAPF, IJTOATCH, IJTOGTF, IJTOHELP, IJTOLSPC, IJTOMEM, IJTOMODU, IJTOMVS, IFTOP, IJTOPOOL, IJTOSET, IJTOSRC, IJTOSTAR, IJTOSTCK, IJTOSTOP, IJTOTASK, IJTOTRAC, IJTOVAVT, IJTOWAIT, TSOAOPER	TSO Command parser
SCAN	UDW1: First four characters of command verb UDW2: Second four characters of command verb UDW3: Halfword length of command, one byte flag one byte reserved UDW4: A(command buffer) for scan UDW5: A(ECB) used during scan Modules: IJTICNFG, IJTOTSO, IJTSCSRB	TSO Command Scanner

## UP

The following is a list of internal trace entries produced by the trace instance initialization routines:

Subtype	User Fields/Modules	Description
UBFE	UDW1: <zero> UDW2: Reason code UDW3: Return code Module: T03STBFF	Error flushing trace buffer.
UFLH	UDW1: <zero> UDW2: <zero> UDW3: Buffer address UDW4: Exit address Module: T01STBFF	Flushing last full trace buffer.
URQT	UDW1: Caller address space ID UDW2: Instance ID UDW3: Group count Module: T03CTSUP	Initial trace instance start request.
USUC	UDW1: Caller address space ID UDW2: Instance ID Module: T03CTSUP	Trace instance start successful.

## WAIT

The following is a list of WAIT-type IFS internal trace entries:

Subtype	User Fields/Modules	Description
	UDW1-UDW5: Not used Modules: IFSSTDRV	Task driver entering wait state.

# API Return Codes

---

This chapter defines diagnostic information returned by the API macro instructions.

For information on API diagnostic codes, refer to the chapter “API Diagnostic Codes.”

This chapter includes:

- [AOPEN and ACLOSE General Return Codes](#)
- [AOPEN and ACLOSE Error Codes](#)
- [TPL-Based General Return Codes – R15](#)
- [Recovery Action Codes – R00](#)
- [Specific Error Codes](#)
- [Exceptional Conditions: RTNCD 04xx](#)
- [Connection and Data Integrity Errors: RTNCD 08xx](#)
- [Execution Environment Errors: RTNCD 0Cxx](#)
- [Format or Specification Errors: RTNCD 10xx](#)
- [Sequence and Procedural Errors: RTNCD 14xx](#)
- [R00 Diagnostic Codes](#)

When a Unicenter TCPaccess API macro instruction completes, the system returns diagnostic information to the application program indicating the success or failure of the request, and if an error occurred, what specific error was encountered.

This chapter presents information for the first two bytes of the return code. To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes.”

## Returned Information

The information presented in this chapter is organized according to the categories of return information. For each return code within a given category, the following information is provided (if appropriate):

- A mnemonic for the return code that is defined in the APCB or TPL dsect
- The actual value of the return code in decimal and hexadecimal format
- A short description of the error
- A detailed description of conditions that may cause the return code to be generated, and in some cases, a suggested course of action

### AOPEN/ACLOSE Macro Instructions

For AOPEN or ACLOSE macro instructions, the following information is returned:

- A general return code returned in register 15 indicating general success or failure of the macro instruction
- An error code returned in register zero, and usually stored in the APCB, indicating the type of error that occurred
- A diagnostic code stored in the APCB that gives specific information about the error



## TPL-Based Macro Instructions

For TPL-based macro instructions, the following information is returned:

- A general return code returned in register 15 indicating general success or failure of the macro instruction.
- The contents of register 00 are dependent on the contents of register 15:

Register 15 Contents	Register 00 Contents
0	Conditional completion code
4	Recovery action code
8	Function code (TPL is not applicable)
12 (or higher)	Four-byte diagnostic code (TPL is not applicable)

- The contents of TPL field TPLRTNCD are also dependent on the contents of register 15:

Register 15 Contents	TPLRTNCD Contents
0	Recovery action code (X'00') and the conditional completion code
4	Recovery action code (X'04' - X'14') and the specific error code
8 (or higher)	Not updated

- Provider-specific and protocol-specific diagnostic codes and sense data that may be helpful in diagnosing the cause of certain errors. The field TPLDGNCD contains a one-byte module ID and a one-byte instance ID. If you are using the extended TPL, the field TPLXDIAG contains a two-byte module ID and a two-byte instance ID. The one-byte module ID in TPLDGNCD can be mapped to the two-byte module ID with the table presented in this chapter.

For more information on how the Assembler API handles errors and special conditions, refer to *Assembler API Concepts*.

## AOPEN and ACLOSE General Return Codes

General return codes are returned in register 15 to indicate the general success or failure of an AOPEN or ACLOSE macro instruction.

If the return code is zero, the operation was successful. If the return code is non-zero, the operation failed. In the latter case, additional information is returned in register zero and the APCB to more specifically identify the error that occurred.

Unlike TPL-based macro instructions, no mnemonics have been defined for AOPEN and ACLOSE return codes. This section lists the general return codes.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

Return Code	Description
0 (X'00')    Operation Successful	<p>The requested operation was successful.</p> <p>If an AOPEN macro was executed, a new transport user has been defined and a session with the API has been established. The APCB has been opened and initialized and can be used to open transport service endpoints.</p> <p>If an ACLOSE macro was executed, the session between the API and the transport user has been terminated, and any endpoints opened by the transport user have been closed. The APCB has been returned to its original state and can be reused in another AOPEN macro instruction.</p>
4 (X'04')    No Operation	<p>No operation was performed (no error code stored in the APCB).</p> <p>If an AOPEN macro instruction was executed, the APCB was already opened.</p> <p>If an ACLOSE macro instruction was executed, the APCB was already closed. The state of the APCB is unchanged, and no operation has been performed. An error code or diagnostic code is not stored in the APCB. However, an error code is returned in register zero.</p>
8 (X'08')    Temporary Failure	<p>A temporary failure occurred. The error code is stored in the APCB.</p> <p>The requested macro instruction failed, but the failure is considered temporary, and the macro instruction may be retried after some delay. The error code returned in register zero is also stored in</p>

Return Code	Description
	<p>the APCB and identifies a particular type of error. The diagnostic code stored in the APCB indicates a specific instance of the error type.</p> <p>This code is returned only by the AOPEN macro instruction; the APCB remains closed.</p>
12 (X'0C') Permanent Failure	<p>A permanent failure occurred. The error code is stored in the APCB.</p> <p>The requested macro instruction failed, and the failure is considered permanent. The error code returned in register zero is also stored in the APCB and identifies a particular type of error.</p> <p>The diagnostic code stored in the APCB indicates a specific instance of the error type.</p> <p>If the macro instruction was AOPEN, the APCB remains closed. The permanent error flag is set in the APCB and must be cleared before another AOPEN macro instruction is attempted. If the macro instruction was ACLOSE, the APCB remains opened. The permanent error flag is not set in the APCB.</p>
16 (X'10') Fatal Error	<p>A fatal error occurred. There is no error code stored in the APCB.</p> <p>A fatal error occurred, and the requested operation could not be performed. An error code is returned in register zero, but is not stored in the APCB. The state of the APCB is unchanged. No diagnostic code is returned in the APCB.</p> <p>A fatal error is generally caused by an invalid APCB address passed in register 1 or an invalid or corrupted APCB.</p>

## AOPEN and ACLOSE Error Codes

If an AOPEN or ACLOSE macro instruction fails, an error code is returned in register zero. Depending on the general return code in register 15, this error code might also be stored in the APCB (at APCBERRC). This error code identifies a particular type of error.

For some types of errors, more than one situation can cause an error to be generated. In this case, if an error code was stored in the APCB, a diagnostic code is also stored (at APCBDGNC) that identifies the specific instance of the error type. This section lists all AOPEN and ACLOSE error codes.

To look up the information for the last two bytes of the return code, see [Converting Two-Byte to Four-Byte Diagnostic Codes](#) (in Chapter5).

For more information about AOPEN error codes, see [C258xxxx \(1Axx\) T012AOPEN](#). For more information about ACLOSE error codes, see [C258xxxx \(1Axx\) T012AOPEN](#).

The following table lists AOPEN and ACLOSE error codes.

Error Code	Description
1 (X'01') APCBECFG	<p>The Unicenter TCPaccess API subsystem has not been configured in the chain of MVS subsystems. Either the subsystem name in the APCB is invalid and the subsystem does not exist, or the name is valid but the subsystem has not been activated since the last IPL of the system.</p> <p>The latter condition can occur if the application program is started before the Unicenter TCPaccess API address space.</p>
2 (X'02') APCBEACT	<p>The Unicenter TCPaccess API subsystem CVT has been located, and the subsystem name in the APCB is probably correct. However, the subsystem has not been activated by the system operator. This can occur if the application program is started before the Unicenter TCPaccess API address space, or if the subsystem has terminated or has been stopped.</p> <p>You can reissue the request after clearing the permanent error flag in the APCB.</p>
3 (X'03') APCBERDY	<p>The Unicenter TCPaccess API subsystem is active, but is in the process of initializing or terminating and cannot service the AOPEN request. The application program should pause shortly and then reissue the AOPEN macro</p>

Error Code	Description
	instruction.
4 (X'04') APCBESTP	<p>The Unicenter TCPaccess API subsystem has been stopped by the system operator, and sessions with new transport users cannot be established.</p> <p>Existing sessions are maintained for a sufficient period to let the application program gracefully terminate its use of the Unicenter TCPaccess API services.</p>
5 (X'05') APCBEDRA	<p>The Unicenter TCPaccess API subsystem is draining (that is, terminating), and no sessions with new transport users can be established.</p> <p>The application program may continue using the API via existing sessions, and when the last session is terminated, the Unicenter TCPaccess API gracefully terminates itself.</p>
6 (X'06') APCBEVCK	<p>An error was encountered while validity checking certain fields within the APCB. If the general return code was 16, the error was fatal, and indicates that the address in register one did not point to a valid APCB, or that the APCB storage area may have been corrupted.</p> <p><b>Note:</b> If the general return code is 12, the error is considered permanent and the permanent error flag is set.</p>
7 (X'07') APCBELER	<p>An internal logic error occurred during the execution of an AOPEN or ACLOSE macro instruction. If the general return code is 16, the error is considered fatal and no diagnostic code is returned. Some possible causes are the inability to establish the top-level ESTAE recovery environment or a system ABEND while attempting to release allocated memory.</p> <p><b>Note:</b> If the general return code is 12, the error is considered permanent and the permanent error flag is set.</p>
8 (X'08') APCBEPRB	<p>An AOPEN or ACLOSE macro instruction was issued and the RB chain contained supervisor or interrupt request blocks. A typical cause of this error is issuing such a macro instruction within an asynchronous exit routine, or within a load module linked to or called by an exit routine.</p> <p>Asynchronous exit routines are preemptive, and run under control of an IRB. The diagnostic code indicates whether an AOPEN or ACLOSE macro instruction was issued.</p>

Error Code	Description
9 (X'09') APCBEOPN	An AOPEN macro instruction was issued and the APCB was already opened. No operation is performed, and the APCB is not modified. No error code or diagnostic code is stored in the APCB.
10 (X'0A') APCBECLS	An ACLOSE macro instruction was issued and the APCB was already closed. No operation is performed, and the APCB is not modified. In particular, no error code or diagnostic code is stored in the APCB.
11 (X'0B') APCBEBSY	An AOPEN or ACLOSE macro instruction was issued while the APCB was busy with another operation. Since AOPEN and ACLOSE run to completion before returning to the application program, the second request must have been issued from another task or from an asynchronous exit associated with the same task. No operation is performed, and the APCB is not modified. In particular, no error code or diagnostic code is stored in the APCB.
12 (X'0C') APCBEPER	<p>An AOPEN macro instruction was issued and the permanent error flag was set in the APCB. No operation is performed, and no information is stored in the APCB.</p> <p>If an AOPEN macro instruction fails with a general return code of 12, the APCB that was intended to be opened has its permanent error flag set. The application program must clear this flag before using the APCB with another AOPEN macro instruction.</p> <p>If an ACLOSE macro instruction is issued with an APCB that has its permanent error flag set, the APCB must already be closed, and APCBECLS is returned as the error code.</p>
13 (X'0D') APCBECVT	<p>An AOPEN macro instruction was issued and the address of the access method's Communications Vector Table (CVT) was zero, indicating that the access method has not completed initialization. This error is similar to APCBERDY, except that it applies to the access method specifically instead of the subsystem in general. The application program should pause momentarily, and then reissue the request.</p> <p>Typically the Unicenter TCPaccess API address space is started during system initialization procedures after an IPL and allowed to stabilize before other application programs are started. Therefore, under normal circumstances, this error should not occur.</p>

Error Code	Description
14 (X'0E') APCBEMEM	Insufficient memory is available within the application program's address space for local data structures allocated and maintained by the Unicenter TCPaccess API. These data structures are allocated from subpool 0. If the current addressing mode is 31-bit and the APCB RMODE is ANY, memory is allocated above 16 MB. Otherwise, memory is allocated below 16 MB. The diagnostic code indicates which data structure was being allocated.
15 (X'0F') APCBEENV	An error occurred while attempting to initialize the language environment during AOPEN processing, or while attempting to terminate the environment during ACLOSE processing.  If the application program is operating in an assembler language environment (ENVIRO=ASM), this error should not occur. Otherwise, the application must be operating within the environment of some higher-level language and should be using the appropriate the Unicenter TCPaccess API runtime library.
16 (X'10') APCBELEG	An error occurred in the PC routine (APICTLEG) that is called to establish a session with the Unicenter TCPaccess API. No session is established, and the APCB is returned to its state before the AOPEN macro instruction was issued. The diagnostic code indicates why the session could not be established.
17 (X'11') APCBEVER	The access method version number stored in the APCB is invalid. If the application program uses the APCB macro instruction to generate the APCB, this error should not occur. If the application program constructs the APCB using some other mechanism, the programmer should make sure the version number is correct. The version number is stored in the low-order nibble with the access method ID (APCBAM). The current version number is zero (0).
18 (X'12') APCBEOPT	An invalid or unsupported option was indicated in the APCB. If the APCB is generated using the APCB macro instruction, this error should not occur. Otherwise, the application program should take care to set all unused bits in the option code (APCBOPTC) to zero.

Error Code	Description
20 (X'14') APCBEAMD	<p>An ACLOSE macro instruction was issued in 24-bit addressing mode for an APCB that was opened in 31-bit mode. If an APCB is opened in 31-bit mode, all subsequent requests associated with the APCB, either directly or indirectly, must be executed in 31-bit mode. An APCB opened in 24-bit mode can be closed in either addressing mode. The application program must make sure that the current addressing mode is consistent with the manner in which the APCB is opened.</p> <p>If the APCB was opened in 24-bit mode and this error indication appears to be erroneous, the APCB may have been corrupted. The programmer should verify that the APCB has not been inadvertently modified by the application program.</p>
21 (X'15') APCBETRV	<p>An ACLOSE macro instruction was issued, and the AM-dependent processing routine could not be invoked. Either the address of the access method transfer vector, which was stored in the APCB by AOPEN, has been modified by the application program, or the transfer vector itself has been corrupted. The latter is less likely because the transfer vector resides in store-protected memory.</p>
22 (X'16') APCBEEND	<p>An error occurred in the PC routine (APICTEND) that is called to terminate a session with the Unicenter TCPaccess API. The session, if it existed, is not terminated, and the APCB remains opened. The diagnostic code stored in the APCB indicates why the session could not be terminated.</p>



## TPL-Based General Return Codes—R15

General return codes are returned in register 15 to indicate the general success or failure of a TPL-based macro instruction. If the macro instruction was executed in synchronous mode, the general return code indicates whether the requested operation was completed normally. If the macro instruction was executed in asynchronous mode, the general return code indicates whether the transport service request was accepted. The subsequent TCHECK macro instruction indicates normal or abnormal completion by returning the same general return code that would have been returned in synchronous mode.

The general return codes defined in this section are generated by the Unicenter TCPaccess API. If a SYNAD or LERAD exit routine was entered because of an error condition, the general return code in register 15 is provided by the exit routine and may differ from the return codes used by the API.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists TPL-based general return codes in R15.

Error Code	Description
0 (X'00') TROKAY	<p>Request accepted, or completed normally or conditionally.</p> <p>If the macro instruction was executed in synchronous mode, the requested operation was completed normally or conditionally without any errors. Register zero contains a conditional completion code that indicates what, if any, unusual conditions occurred.</p> <p>If the macro instruction was executed in asynchronous mode, the request was accepted and the requested operation was initiated.</p> <p><b>Note:</b> A TCHECK macro instruction is required to obtain the completion status of the request.</p>
4 (X'04') TRFAILED	<p>Request not accepted, or request completed abnormally due to a non-recoverable error or abnormal condition.</p> <p>If the macro instruction was executed in synchronous mode, the operation was completed abnormally. Register zero contains a recovery action code unless this information has been modified by the SYNAD or LERAD exit routine.</p> <p>If the macro instruction was executed in asynchronous mode, the request was not accepted because of an error or abnormal condition. Register zero contains a recovery action code unless this information has been modified by the SYNAD or LERAD exit routine.</p>

Error Code	Description
	<b>Note:</b> A TCHECK macro instruction is not required because the requested operation was never initiated.
8 (X'08') TRFATLFC	<p>Requested operation could not be initiated because of an invalid function code.</p> <p>A fatal error occurred due to an invalid function code. The requested operation could not be initiated and the invalid function code was returned in register zero. The SYNAD and LERAD exit routines are not entered.</p> <p>If the macro instruction was executed in asynchronous mode, a TCHECK macro instruction should not be executed.</p>
12 (X'0C') TRFATLPL	<p>Requested operation could not be initiated because of an invalid or corrupted TPL.</p> <p>A fatal error occurred due to an invalid or corrupted TPL. Either the TPL address was incorrect or the TPL did not contain a valid control block identifier. If the TPL appears to be correct, then it may have been located in store-protected memory. The requested operation could not be initiated and a diagnostic code is returned in register zero. The SYNAD and LERAD exit routines are not entered.</p> <p><b>Note:</b> If the macro instruction was executed in asynchronous mode, a TCHECK macro instruction should not be executed.</p>
16 (X'10') TRFATLAM	<p>Requested operation could not be initiated because of a fatal access method error.</p> <p>A fatal error occurred due to an internal access method error. Most likely an internal control block has been corrupted or information stored in the APCB by the Unicenter TCPaccess API has been inadvertently changed. The requested operation could not be initiated and a diagnostic code is returned in register zero. The SYNAD and LERAD exit routines are not entered.</p> <p><b>Note:</b> If the macro instruction was executed in asynchronous mode, a TCHECK macro instruction should not be executed.</p>
20 (X'14') TRFATLAP	<p>Requested operation could not be initiated because the APCB is closed.</p> <p>A fatal error occurred due to a closed or corrupted APCB. The requested operation could not be initiated and a diagnostic code is returned in register zero. The SYNAD and LERAD exit routines are not entered.</p> <p><b>Note:</b> If the macro instruction was executed in asynchronous mode, a TCHECK macro instruction should not be executed.</p>
24 (X'18')	First user return code from SYNAD or LERAD exit routine.

Error Code	Description
TRUSER	This return code is defined to provide a reference for application programs that want to assign application-specific return codes generated within a SYNAD or LERAD exit routine. General return codes equal to or greater than this value does not conflict with those used by the Unicenter TCPaccess API.

## Recovery Action Codes—R00

The recovery action code is stored in the TPL return code field (TPLRTNCD) and returned in register zero whenever the general return code in register 15=X'04'. The recovery action code defines categories of errors that are handled in a common manner and can serve as an index into a branch table to determine processing action within an error routine.

The recovery action codes defined in this section are those generated by the Unicenter TCPaccess API. If SYNAD or LERAD exit routines are being used, this code is also passed to the exit routine in register zero. If the exit routine does not modify register zero, the recovery action code is returned to the application program.

To look up the information for the last two bytes of the return code, see the chapter "API Diagnostic Codes".

The following table lists recovery action codes.

Error Code	Description
0 (X'00') TAOKAY	Request accepted, or completed normally or conditionally. This is the value initially set in the TPL return code field. If this value remains after request acceptance or completion, an error did not occur. In this case, the conditional return code (also stored in the TPL return code field) is returned to the application program in register zero.
4 (X'04') TAEXCPTN	An exceptional condition occurred that prevented normal completion or acceptance of the requested function. <b>Note:</b> Errors of this type generally do not affect the integrity of a network connection or the data transferred over it.
8 (X'08') TAINTEG	Connection or data integrity error. An error or abnormal condition occurred that can affect the integrity of the

Error Code	Description
	connection associated with an endpoint or the data received or transmitted over it.
12 (X'0C') TAENVIRO	Failure due to abnormal environmental condition. This code is generally indicative of an abnormal condition in the execution environment that is outside the direct control of the application program. Some external action may be required to relieve the condition.
16 (X'10') TAFORMAT	A failure occurred because of a format or specification error, usually associated with a parameter provided by the application program in a TPL-based request.  Errors of this type generally indicate a logic error in the application program and should not occur once the program is debugged.
20 (X'14') TAPROCED	A failure occurred because a request was issued out of sequence or when the endpoint was in an inappropriate state, or a procedural requirement was violated.  Errors of this type generally indicate a logic error in the application program and should not occur once the program is debugged.
24 (X'18') TATPLERR	A logic error occurred, but the TPL associated with the request was in a state or condition that prevents storing the recovery action code and specific error code in the RTNCD field of the TPL.  The recovery action code is returned to the application program in register zero as usual, but no specific error code is available.
28 (X'1C') TAUSER	This recovery action code is defined to provide a reference for application programs that want to assign application-specific return codes generated by a SYNAD or LERAD exit routine. Recovery action codes equal to or greater than this value does not conflict with those used by the Unicenter TCPaccess API.

## Conditional Completion Codes—R00

Conditional completion codes are returned in register 0 whenever the general return code in register 15 indicates normal or conditional completion. The conditional completion code is also stored in the TPL return code field (TPLRTNCD) in place of the specific error code (TPLERRCD). The recovery action code (TPLACTCD) is set to zero (TAOKAY) to indicate normal or conditional completion.

Conditional completion codes are used to indicate unusual conditions that accompanied an otherwise normal completion of the requested operation. These codes typically indicate the occurrence of a condition that normally does not affect the successful execution of the application program and should not be treated as an error, or special completion status the transport provider needs to return to the application program.

Each bit of the eight-bit completion code represents a particular condition. Therefore, unlike specific error codes, a conditional completion code can represent the presence of more than one condition.

To look up the information for the last two bytes of the return code, see the section “Converting Two-Byte to Four-Byte Diagnostic Codes” in the chapter “API Diagnostic Codes”.

The following table lists conditional completion codes.

Error Code	Description
0 (X'00) TCOKAY	The requested operation completed normally without any unusual conditions. The requested operation was performed, and all appropriate information was returned.
4 (X'04) TCTIME	A TRECVC completed based on a timeout expiring. The TPL is updated to reflect the length of data received so far.
8 (X'08) TCSTOP	The system operator entered a command from the operator's console to stop the Unicenter TCPaccess API subsystem or the transport provider. Graceful shutdown procedures have begun, but the application program is allowed to continue operation for a limited amount of time. The application program should begin a graceful shutdown of the endpoint.
32 (X'20) TCTRUNC	One or more return values would not fit in the storage area provided by the application program and OPTCD=TRUNC was specified. As much information as would fit in the storage area is returned, and the remainder is discarded.

Error Code	Description
64 (X'40') TCNEGOT	One or more options or user-supplied values were negotiated to an inferior value because they were beyond the supported range. OPTCD=NEGOT was also set in TPLOPTCD. This code applies only when the option or RTNCD 00xx facility is supported but the value is out of range.
128 (X'80') TCVERIFY	The requested option is not supported. One or more options did not verify with the TOPTION service request.

## R00 Diagnostic Codes

When the general return code in R15=X'12' or higher, Register 00 contains the four-byte diagnostic codes described in [API Diagnostic Codes](#) (Chapter 5). In this case, the TPL is not updated.

## Specific Error Codes

When the general return code is X'04' (TRFAILED) and the recovery action code is less than X'24' (TATPLERR), a specific error code is stored with the recovery action code in the TPL return code field (TPLRTNCD). This error code provides a more definitive characterization of the failure and can be used in combination with the recovery action code to determine the precise error recovery procedure.

## Recovery Action Code Combinations

The following table shows the valid combinations of recovery action codes and specific error codes. An entry in the table indicates that the column and row values are valid combinations for TPLRTNCD. The label appearing in the table is the mnemonic defined in the TPL DSECT.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

Value	TAEXCPTN X'04'	TAINTEG X'08'	TAENVIRO X'0C'	TAFORMAT X'10'	TAPROCED X'14'	TATPLERR 0x'18'
X'00'						
X'01'		TEPROTO	TESYSERR	TEBDOPCD	TESTATE	TEB4EXIT
X'02'		TEOVRFLO	TESUBSYS	TEBDEPID	TEINEXIT	TEACTIVE
X'03'		TEDISCON	TENOTCNF	TEBDXECB	TEINACTV	
X'04'		TERELESE	TENOTACT	TEBDDOM	TEINCMPL	
X'05'		TEOVLAY	TENOTRDY	TEBDPROT	TEINDICA	
X'06'	TENONEGO		TEDRAIN	TEBDTYPE	TEBUFOVR	
X'07'			TESTOP	TEBDXLST	TEBEQOVR	
X'08'			TETERM	TEBDUSER	TENOCONN	
X'09'	TENOBLOK	TEFLOW	TEUNSUPO	TEBDACEE	TENODISC	
X'0A'	TENOLSTN	TERETRCT	TEUNSUPF	TEBDSQNO	TEOUTSEQ	
X'0B'		TEPURGED	TEUNAVBL	TEBDQLEN	TENOERR	
X'0C'			TEUNAETH	TEBDTCB		
X'0D'			TERSOURC	TEBDASCB	TEAMODE	
X'0E'			TEINUSE	TEBDADDR	TEOWNER	
X'0F'			TEUSRXIT	TEBDOPTN	TELISTEN	
X'10'				TEBDATA	TEACCEPT	
X'12'				TEBDTSID		

## Exceptional Conditions: RTNCD 04xx

This section includes the exceptional condition specific error codes.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists exceptional condition error codes.

Hex Code	Description
6 (X'06') TENONEGO	One or more options or values were out of range and could not be negotiated because OPTCD=NONEGOT was specified (see <a href="#">Conditional Completion Codes - R00</a> ).
9 (X'09') TENOBLOK	No blocking allowed. A request was issued with OPTCD=NOBLOCK specified and the request could not be completed immediately.
10 (X'0A') TENOLSTN	A TRETRACT was issued and no TLISTEN was outstanding.



## Connection and Data Integrity Errors: RTNCD 08xx

This section lists the connection and data integrity error specific error codes.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists connection and data integrity errors.

Hex Error Code	Description
1 (X'01') TEPROTO	A protocol error occurred that prevents successful completion of the request.
2 (X'02') TEOVRFLO	The data to be returned to the user does not fit into the storage area provided and OPTCD=NOTRUNC was specified. In this case, no data is returned to the user and the buffer length is not changed.
3 (X'03') TEDISCON	The endpoint was disconnected, either remotely or locally (for example, by the operator).
4 (X'04') TERELESE	An orderly release was received from the remote TU.
5 (X'05') TEOVLAY	A control block was overlaid.
9 (X'09') TEFLOW	The request could not be completed because of a temporary flow control condition. The request can be retried later after the flow condition is relieved. In the case of the XWA pool, it may need to be increased because it cannot expand dynamically.
10 (X'0A') TERETRCT	An outstanding TLISTEN request was retracted with TRETRACT. This error code applies only to TLISTEN.
11 (X'0B') TEPURGED	A pending request was purged by TCLOSE due to the endpoint becoming disconnected or closed.

## Execution Environment Errors: RTNCD 0Cxx

This section includes the execution environment specific error codes.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists execution environment errors.

Error Code	Description
1 (X'01') TESISERR	An MVS system ABEND occurred within the Unicenter TCPaccess API or the transport provider that is not covered by another error code.
2 (X'02') TESUBSYS	An MVS subsystem error (other than an ABEND) occurred within the Unicenter TCPaccess API for which a more appropriate error code does not exist.
3 (X'03') TENOTCNF	The subsystem is not configured in MVS. The subsystem is not installed (that is, no SSCVT found with matching subsystem name).
4 (X'04') TENOTACT	The subsystem is installed, but not started or active (that is, SSCTUSE is zero).
5 (X'05') TENOTRDY	The subsystem is installed, is active (SSCTSUSE is nonzero), but has not completed initialization or is not accepting requests from external address space.
6 (X'06') TEDRAIN	The system operator entered a command to drain the Unicenter TCPaccess API subsystem. Graceful shutdown procedures have begun, but the application program is allowed to continue operation. The application program should begin a graceful shutdown of the endpoint.
7 (X'07') TESTOP	The subsystem or TP was stopped by the operator (graceful halt).
8 (X'08') TETERM	The subsystem or TP has terminated abnormally.
9 (X'09') TEUNSUPO	The requested option or facility(suppressible) is not supported and OPTCD=UNCOND was specified.

Error Code		Description
10 (X'0A')	TEUNSUPF	The requested function or service (non-suppressible) is not supported.  TEUNSUPF is intended to apply to a particular T-function or service in its entirety, whereas TEUNSUPO applies to a particular nuance of a specific function that is supported.
11 (X'0B')	TEUNAVBL	The requested service or facility is unavailable (for example, offline) at this time.
12 (X'0C')	TEUNAUTH	The requested service or facility is available, but the application program or user is not authorized to use it.
13 (X'0D')	TERSOURC	The request could not be completed because certain required resources were not available.
14 (X'0E')	TEINUSE	A requested resource or facility is serially reusable and is allocated to another user at the time requested.
15 (X'0F')	TEUSRXIT	An API request was failed by a user exit routine.

## Format or Specification Errors: RTNCD 10xx

The error code TEBDxxxx indicates an error in the format or specification of the xxxx parameter. These are some examples:

In general, the Unicenter TCPaccess API interface routine determines:

- The parameter is required but was not specified (the parameter length determines whether or not the parameter is specified).
- The parameter length is non-zero but the parameter address is zero
- The parameter address is invalid (for example, inappropriate storage key, nonexistent storage, and so forth)
- The Unicenter TCPaccess API PC routine determines: The parameter length is out of range (for example, too short or too long)
- The transport provider determines: The parameter length and address are valid, but the content or format is no

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists format or specification errors.

Error Code	Description
1 (X'01') TEBDOPCD	Invalid option code.
2 (X'02') TEBDEPID	Invalid endpoint ID.
3 (X'03') TEBDXECB	Invalid ECB/EXIT address.
4 (X'04') TEBDDOM	Invalid communication domain.
5 (X'05') TEBDPROT	Invalid transport protocol number.
6 (X'06') TEBDTYPE	Invalid transport service type.
7 (X'07') TEBDXLST	Invalid exit list.
8 (X'08') TEBDUSER	Invalid user block or user block address.
9 (X'09') TEBDACEE	Invalid ACEE or ACEE address.
10 (X'0A') TEBDSQNO	Invalid sequence number.
11 (X'0B') TEBDQLEN	Invalid queue length parameter.
12 (X'0C') TEBDTCB	Invalid TCB or TCB address.
13 (X'0D') TEBDASCB	Invalid ASCB or ASCB address.
14 (X'0E') TEBDADDR	Invalid protocol address.

15 (X'0F')	TEBDOPTN	Invalid protocol option.
16 (X'10')	TEBDDATA	Invalid user data parameter.
18 (X'12')	TEBDTSID	Invalid transport service ID.

## Sequence and Procedural Errors: RTNCD 14xx

This subsection includes sequence and procedural error codes.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists sequence and procedural errors.

Error Code	Description
1 (X'01') TESTATE	The current state of the endpoint is invalid for the requested function. Because the endpoint state is known only in the user's address space, only the Unicenter TCPaccess API interface routine can generate this error code.
2 (X'02') TEINEXIT	A function that cannot be executed within an exit routine was requested.
3 (X'03') TEINACTV	A TCHECK was issued to an inactive TPL.
4 (X'04') TEINCMPL	A request was issued on an endpoint that had pending requests (for example, an incomplete function request). The request could not be executed because a previous request has not yet completed.
5 (X'05') TEINDICA	The request could not be completed because of a pending connect indication.
6 (X'06') TEBUFOVR	The Send/Receive buffer is overrun. The cumulative amount of space required for incomplete TSEND/TSENDTO exceeds the allocated limit.
7 (X'07') TEREQOVR	The Send/Receive request is overrun. The total number of pending TSEND/TSENDTO or TRECVR/TRECVR requests exceeds the allocated limit.
8 (X'08') TENOCONN	No connect is indication.

Error Code	Description
9 (X'09') TENODISC	No disconnect indication is pending.
10 (X'0A') TEOUTSEQ	A request was issued in a sequence that is not supported. A TOPTION request was issued with OPTCD=API specifying the SRE of a buffer when the buffer has already been allocated. Buffers are allocated by data transfer requests.
11 (X'0B') TENOERR	No datagram error indication is pending.
13 (X'0D') TEAMODE	The current addressing mode (AMODE) is inconsistent with the addressing mode at the time the APCB was opened.
14 (X'0E') TEOWNER	The request issued must be issued by the task that opened the endpoint for example, TCLOSE, TACCEPT).
15 (X'0F') TELISTEN	A TLISTEN was issued when the current number of outstanding connect indications was at its maximum value.
16 (X'10') TEACCEPT	The endpoint is busy due to a TACCEPT being issued specifying this endpoint as the new endpoint.

## Logic Errors with No TPL Return Code: 18xx

This subsection includes the logic errors with no TPL return code.

To look up the information for the last two bytes of the return code, see the chapter “API Diagnostic Codes”.

The following table lists logic errors with no TPL return code.

Error Code		Description
1 (X'01')	TEB4EXIT	A TCHECK macro instruction was executed using a TPL marked incomplete but specified a TPL exit routine. The TCHECK macro instruction was apparently issued before the exit routine was entered. No information is stored in the TPLRTNCD field.
2 (X'02')	TEACTIVE	A TPL-based macro instruction (other than TCHECK) was executed using an active TPL. No information is stored in the TPLRTNCD field.





# API Diagnostic Codes

---

This chapter defines diagnostic information returned by the Unicenter TCPaccess API macro instructions. It includes tables, listed by module name, with both two-byte and four-byte diagnostic codes and their descriptions. For more information on API codes returned in registers, refer to the chapter “API Return Codes.” The following diagnostic code ranges are included in this chapter:

- [C006xxxx T010PNW](#)
- [C100xxxx T010PPC](#)
- [C200xxxx T010SPC](#)
- [C301xxxx \(24xx\) T01XACPT](#)
- [C404xxxx \(3Bxx\) T01ASWDN](#)
- [C502xxxxT01ESCF](#)
- [C602xxxx T01SMOVE](#)
- [C704xxxx T01AMIUC](#)
- [C805xxxx T01XTTCP](#)
- [C901xxxx T01SIMUX](#)

## Diagnostic Codes

Two bytes of diagnostic codes are stored for certain errors to provide more information about their cause. Often this information is protocol- or implementation-dependent and is not standardized across all transport providers. Therefore, the application program should use this data for informational purposes only and should not make procedural decisions based on its content. It is recommended that any diagnostic messages written by the application program in response to an error contain a copy of this information.

If the high order bit of the diagnostic code is on, the code is the MVS system ABEND code. For these codes, refer to the appropriate IBM manual.

If the high order bit is off, the first byte is the module ID number and the second byte is a sequential error number assigned by the module. This table identifies the source module that issues a particular diagnostic code.

If you are using the expanded TPL, diagnostic codes are four bytes: two bytes for the module ID and two bytes for the instance codes.

Short (two-byte) diagnostic codes are returned if you are using a TPL that does not have the expanded diagnostic field. The TPL field TPLDGNCD holds a one-byte module ID and a one-byte instance ID. These short codes are then mapped to a four-byte diagnostic code. If you use a TPL that has the expanded diagnostic field, the full four-byte code is returned in field TPLXDIAG. The section [Converting Two-Byte to Four-Byte Diagnostic Codes](#) gives a table for converting the codes.

The tables list both the four-byte code and the two-byte code. The instance ID uses only the last byte of the two-byte instance field in TPLXDIAG so no mapping is required for the one-byte instance ID.

As previously described, you will need to refer to the values returned in Register 0 and Register 15 to determine the type of error that occurred.

The first byte of the two-byte diagnostic code represents the module that issued the code. The second byte specifies the instance ID that indicates where in the module the error occurred.

## Converting Two-Byte to Four-Byte Diagnostic Codes

Use the following table to convert two-byte diagnostic codes to their four-byte representation. The table also gives a cross reference to the table describing the codes.

The following table lists two-byte to four-byte diagnostic code conversion.

Two-Byte Code Range	Four-Byte Code and Module
0100 - 01FF	<a href="#">C240xxxx (01xx) T012SPC</a>
0300 - 03FF	<a href="#">C241xxxx (03xx) T012TACC</a>
0400 - 04FF	<a href="#">C242xxxx (04xx) T012TADR</a>
0500 - 05FF	<a href="#">C243xxxx (05xx) T012TBIN</a>
0600 - 06FF	<a href="#">C244xxxx (06xx) T012TCLR</a>
0700 - 07FF	<a href="#">C245xxxx (07xx) T012TCLS</a>
0800 - 08FF	<a href="#">C246xxxx (08xx) T012TCNF</a>
0900 - 09FF	<a href="#">C247xxxx (09xx) T012TCON</a>
0A00 - 0AFF	<a href="#">C248xxxx (0Axx) T012TDIS</a>
0B00 - 0BFF	<a href="#">C249xxxx (0Bxx) T012TINF</a>
0C00 - 0CFF	<a href="#">C24Axxxx (0Cxx) T012TLIS</a>
0D00 - 0DFF	<a href="#">C24Bxxxx (0Dxx) T012TOPN</a>
0E00 - 0EFF	<a href="#">C24Cxxxx (0Exx) T012TOPT</a>
0F00 - 0FFF	<a href="#">C24Dxxxx (0Fxx) T012TRCV</a>
1000 - 10FF	<a href="#">C24Exxxx (10xx) T012TRER</a>
1100 - 11FF	<a href="#">C24Fxxxx (11xx) T012TRFR</a>
1200 - 12FF	<a href="#">C250xxxx (12xx) T012TRJT</a>
1300 - 13FF	<a href="#">C251xxxx (13xx) T012TRLK</a>
1400 - 14FF	<a href="#">C252xxxx (14xx) T012TRLS</a>
1500 - 15FF	<a href="#">C253xxxx (15xx) T012TRCT</a>
1600 - 16FF	<a href="#">C254xxxx (16xx) T012TSND</a>
1700 - 17FF	<a href="#">C255xxxx (17xx) T012TSTO</a>
1800 - 18FF	<a href="#">C256xxxx (18xx) T012TUNB</a>
1900 - 19FF	<a href="#">C257xxxx (19xx) T012TUSR</a>

Two-Byte Code Range	Four-Byte Code and Module
1A00 - 1AFF	<a href="#"><u>C258xxxx (1Axx) T012AOPN</u></a>
1B00 - 1BFF	<a href="#"><u>C259xxxx (1Bxx) T012ACLS</u></a>
1C00 - 1CFF	<a href="#"><u>C25Axxxx (1Cxx) T012TCHK</u></a>
1D00 - 1DFF	<a href="#"><u>C25Bxxxx (1Dxx) T012TERR</u></a>
1E00 - 1EFF	<a href="#"><u>C25Cxxxx (1Exx) T012TSTA</u></a>
2100 - 21FF	<a href="#"><u>C25Fxxxx (21xx) T01PSTUB</u></a>
2400 - 24FF	<a href="#"><u>C301xxxx (24xx) T01XACPT</u></a>
2500 - 25FF	<a href="#"><u>C302xxxx (25xx) T01XBIND</u></a>
2600 - 26FF	<a href="#"><u>C303xxxx (26xx) T01XCLS1</u></a>
2700 - 27FF	<a href="#"><u>C304xxxx (27xx) T01XCLS2</u></a>
2800 - 28FF	<a href="#"><u>C305xxxx (28xx) T01XCONN</u></a>
2900 - 29FF	<a href="#"><u>C306xxxx (29xx) T01XUBN</u></a>
2A00 - 2AFF	<a href="#"><u>C307xxxx (2Axx) T01XCREA</u></a>
2B00 - 2BFF	<a href="#"><u>C308xxxx (2Bxx) T01XFREE</u></a>
2D00 - 2DFF	<a href="#"><u>C30Cxxxx (2Dxx) T01XLIST</u></a>
2E00 - 2EFF	<a href="#"><u>C30Dxxxx (2Exx) T01XOPT1</u></a>
2F00 - 2FFF	<a href="#"><u>C30Exxxxx (2Fxx) T01XPEER</u></a>
3001 - 3007	<a href="#"><u>C30Fxxxx (30xx) T01XPRTA</u></a>
3100 - 31FF	<a href="#"><u>C311xxxx (31xx) T01XREAD</u></a>
3300 - 33FF	<a href="#"><u>C315xxxx (33xx) T01XSHT1</u></a>
3400 - 34FF	<a href="#"><u>C316xxxx (34xx) T01XSHT2</u></a>
3500 - 35FF	<a href="#"><u>C317xxxx (35xx) T01XSND1</u></a>
3600 - 36FF	<a href="#"><u>C318xxxx (36xx) T01XSND2</u></a>
3B00 - 3BFF	<a href="#"><u>C404xxxx (3Bxx) T01ASWDN</u></a>
3700 - 37FF	<a href="#"><u>C4FFxxxx (37xx) T01ASFRR</u></a>
4300 - 43FF	<a href="#"><u>C526xxxx (43xx) T01ETDI</u></a>
4500 - 45FF	<a href="#"><u>C528xxxx (45xx) T01ETTP</u></a>
5B00 - 5BFF	<a href="#"><u>C932xxxx (5Bxx) T01SRSND</u></a>
5C00 - 5CFF	<a href="#"><u>C933xxxx (5Cxx) T01SRUNB</u></a>
5D00 - 5DFF	<a href="#"><u>C940xxxx (5Dxx) T01STCLO</u></a>

Two-Byte Code Range	Four-Byte Code and Module
5E00 - 5EFF	<a href="#">C941xxxx (5Exx) T01STCON</a>
5F00 - 5FFF	<a href="#">C942xxxx (5Fxx) T01STFIN</a>
6000 - 60FF	<a href="#">C943xxxx (60xx) T01STIMU</a>
6100 - 61FF	<a href="#">C944xxxx (61xx) T01STIN</a>
6200 - 62FF	<a href="#">C945xxxx (62xx) T01STLIS</a>
6300 - 63FF	<a href="#">C946xxxx (63xx) T01STREA</a>
6400 - 64FF	<a href="#">C947xxxx (64xx) T01STRST</a>
6500 - 65FF	<a href="#">C948xxxx (65xx) T01STSND</a>
6900 - 69FF	<a href="#">C94Cxxxx (69xx) T01STTMK</a>
6A00 - 6AFF	<a href="#">C94Dxxxx (6Axx) T01STTML</a>
6B00 - 6BFF	<a href="#">C94Exxxx (6Bxx) T01STTMP</a>
6C00 - 6CFF	<a href="#">C94Fxxxx (6Cxx) T01STTMR</a>
6D00 - 6DFF	<a href="#">C950xxxx (6Dxx) T01STTMT</a>
6E00 - 6EFF	<a href="#">C951xxxx (6Exx) T01STUNB</a>
6F00 - 6FFF	<a href="#">C262xxxx (6Fxx) T012TPLK</a>

## C006xxxx T010PNW

The following table lists X'C006' instance codes.

Four-Byte Code	Description
C0060001	Domain not supported (not AF_INET); message number T01OE111E is generated (described in <i>Prefixed Messages</i> ).
C0060002	Insufficient storage for Inode creation; message number T01OE112E is generated (described in <i>Prefixed Messages</i> ).
C0060003	A serious error was encountered while invoking IEANTCR services for Name/Token Pair; message number T01OE114E is generated (described in <i>Prefixed Messages</i> ).

## C100xxxx T010PPC

The following table lists X'C100' instance codes.

Four-Byte Code	Description
C1000001	Unicenter TCPaccess not up; resources not available (subsystem ID not found).
C1000002	Unicenter TCPaccess not up; resources not available (SSIT ASID=0).
C1000003	Unicenter TCPaccess not up; resources not available (ASCB is equal to or less than zero).
C1000004	Unicenter TCPaccess not up; resources not available (ASCB not matched)
C1000005	Unicenter TCPaccess not up; resources not available (SAVT=0)
C1000006	Unicenter TCPaccess not up; resources not available (SAVX=0).
C1000007	Unicenter TCPaccess not up; resources not available (no program call number).
C1000008	Unicenter TCPaccess shutdown in progress (SSIT ASID=0).
C1000009	Unicenter TCPaccess shutdown in progress (ASCB=<0).
C100000A	Unicenter TCPaccess shutdown in progress (ASCB not matched).
C100000B	Unicenter TCPaccess shutdown in progress (SAVT=0).
C100000C	Unicenter TCPaccess shutdown in progress (SAVX=0).
C100000D	Unicenter TCPaccess shutdown in progress (no program call number).

## C101xxxx T010PSK

The following table lists X'C101' instance codes.

Four-Byte Code	Description
C1010001	Not a socket call; socketpair call is not supported.
C1010002	Too many files (sockets) open. The maximum number of supported sockets has been reached. Increase the number of sockets supported by OpenEdition and rerun the job.
C10100xx	Unable to get vnode; <i>xx</i> is the return code from Osi_Getvnode. Return codes from individual calls are documented in IBM's <i>OpenEdition MVS File System Interface Reference</i> .

## C102xxxx T010PAC

The following table lists X'C102' instance codes.

Four-Byte Code	Description
C1020001	Too many files (sockets) open. The maximum number of supported sockets has been reached. Increase the number of sockets supported by OpenEdition and rerun the job.
C10200xx	Unable to get vnode; <i>xx</i> is the return code from Osi_Getvnode. Return codes from individual calls are documented in the IBM manual OpenEdition MVS File System Interface Reference.

## C107xxxx T010PSO

The following table lists X'C107' instance codes.

Four-Byte Code	Description
C1070001	Socket direction is not a get or set. The getsockopt or setsockopt function is expected, but the value passed in the OpenEdition parameter list is invalid.

## C200xxxx T010SPC

The following table lists X'C200' instance codes.

Four-Byte Code	Description
C2000001	Storage unavailable.
C2000002	SPCB not found.
C2000003	PSVT not found.
C2000004	SEPM address not provided.
C2000005	SEPM ID is incorrect.
C2000006	SEPM ASID/ASCB/TCB is incorrect.
C2000007	SEPM INOD is incorrect
C2000008	Endpoint terminating.
C2000009	Caller has active FRR.
C200000A	Unicenter TCPaccess shutdown in progress (SAVT=0).
C200000B	Unicenter TCPaccess shutdown in progress (SAVX=0).
C200000C	Caller's ASID is out of range.
C200000D	Unable to obtain storage for CSQB.
C200000E	Unicenter TCPaccess is terminating.
C200000F	Unicenter TCPaccess was recycled.
C20000FF	User ABEND. This module abended while processing a socket interface request. Ensure that the parameters to this call are correct.



## C201xxxx T01OSSK

The following table lists X'C201' instance codes.

Four-Byte Code	Description
C2010001	Memory not sufficient to get SEPM; sockets endpoint control block cannot be built due to storage constraints.
C2010002	Socket type not supported; greater than 255.
C2010003	Socket type not supported; not SOCK_STREAM, SOCK_DGRAM or SOCK_RAW.
C2010004	Socket protocol not supported, greater than 255.
C2010005	Socket protocol not supported; not compatible with socket type.
C2010006	Socket protocol not supported; not compatible with socket type.
C2010007	Socket protocol not supported; not compatible with socket type.
C2010008	Socket refused, Unicenter TCPaccess is shutting down.
C201000A	Insufficient memory to get MBUF; memory buffer area cannot be built due to storage constraints.
C201000B	Not privileged for this call (type=SOCK_RAW); requires superuser/system privilege.
C201000C	A shortage exists in the Unicenter TCPaccess address space. New socket creation is denied.
C201xxxx	Access error from SAF security routine service call; xxxx is SAFRC/RC/RSN code.

## C202xxxx T010SAC

The following table lists X'C202' instance codes.

Four-Byte Code	Description
C2020001	Osi_structure size incorrect; control block length value passed in parameter list is outside the minimum/maximum range allowed.
C2020002	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C2020003	Accept refused, Unicenter TCPaccess is shutting down.
C20201xx	Osi_wait call failed; event code is SETUPSIG; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual OpenEdition MVS File System Interface Reference.
C20202xx	Osi_wait call failed; event code is SUSPEND; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual OpenEdition MVS File System Interface Reference.
C20203xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20204xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20205xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20206xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20207xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20208xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20209xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C2020999	Osi_wait function unavailable; invalid address for OSI services(OSIT)
C2020Axx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.

Four-Byte Code	Description
C2020Bxx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C2020Cxx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C2020Dxx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C2020Exx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.

## C203xxxx T010SBD

The following table lists X'C203' instance codes.

Four-Byte Code	Description
C2030002	Sockaddr structure size is incorrect; a minimum length of eight is required; the value passed was less than minimum.
C2030004	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C2030008	Insufficient memory to get MBUF; memory buffer area cannot be built due to storage constraints.
C2030009	Insufficient authority; not system or superuser
C20301xx	Osi_wait call failed; event code is SETUP; xx is return code from Osi_wait.  Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C20302xx	Osi_wait call failed; event code is SUSPEND; xx is return code from Osi_wait.  Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C203xxxx	Access error from SAF security routine service call; xxxx is SAFRC/RC/RSN code. If the high order bit is on in this reason code, See <a href="#">Security Access Errors</a> for more information.

## C204xxxx T010SCL

The following table lists X'C204' instance codes.

Four-Byte Code	Description
C2040001	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20403xx	Get Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20404xx	Get Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20405xx	Free Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20406xx	Free Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20407xx	Free Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20408xx	Free Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20409xx	Get Latch failed; <i>xx</i> is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.

## C205xxxx T010SCN

The following table lists X'C205' instance codes.

Four-Byte Code	Description
C2050001	Sockaddr structure size incorrect. A minimum length of eight is required; the value passed was less than minimum.
C2050002	Osi_structure size incorrect; control block length value passed in parameter list is outside the minimum/maximum range allowed.
C2050003	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20501xx	Osi_wait call failed; event code is SETUPSIG; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C20502xx	Osi_wait call failed; event code is SUSPEND; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C20503xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20504xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20505xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20506xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20507xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20508xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C2050999	Osi_wait function unavailable; invalid address for OSI services (OSIT).

## C206xxxx T010SGN

The following table lists X'C206' instance codes.

Four-Byte Code	Description
C2060001	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20601xx	Get Latch failed; <i>xx</i> is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20602xx	Free Latch failed; <i>xx</i> is llatch return code. See <a href="#">llatch Errors</a> for more information.

## C207xxxx T010SGO

The following table lists X'C207' instance codes.

Four-Byte Code	Description
C2070001	Option data length is zero; invalid value passed in OpenEdition parameter list for option buffer size from getsockopt call.
C2070002	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20701xx	Get Latch failed; <i>xx</i> is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20702xx	Free Latch failed; <i>xx</i> is llatch return code. See <a href="#">llatch Errors</a> for more information.

## C208xxxx T010SSO

The following table lists X'C208' instance codes.

Four-Byte Code	Description
C2080001	Osi_structure size incorrect; control block length value passed in parameter list is outside the minimum/maximum range allowed.
C2080002	Option data length is invalid; unexpected value passed in OpenEdition parameter list for option buffer size from setsockopt call.
C2080003	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20801xx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20802xx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.

## C209xxxx T010SIO

The following table lists X'C209' instance codes.

Four-Byte Code	Description
C2090002	Osi_structure size incorrect; control block length value passed in parameter list is outside the minimum/maximum range allowed.
C2090003	Invalid argument length; unexpected value passed in OpenEdition parameter list.
C2090004	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C2090005	Insufficient memory to get MBUF; memory buffer area cannot be built due to storage constraints.
C2090006	Osi_wait function unavailable; invalid address for OSI services (OSIT).
C2090008	Insufficient memory to get MBUF; memory buffer area cannot be built due to storage constraints.

Four-Byte Code	Description
C20901xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20902xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C209xxxx	Access error from SAF security routine service call; xxxx is SAFRC/RC/RSN code. If the high order bit is on in this reason code, See <a href="#">Security Access Errors</a> later in this chapter to determine the return codes.

## C20Axxxx T010SLI

The following table lists X'C20A' instance codes.

Four-Byte Code	Description
C20A0001	Socket type not set to stream; stream type socket required for listen().
C20A0003	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20A01xx	Get Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.
C20A02xx	Free Latch failed; xx is llatch return code. See <a href="#">llatch Errors</a> for more information.



## C20Bxxxx T010SRD

The following table lists X'C20B' instance codes.

Four-Byte Code	Description
C20B0001	Read byte count passed in OpenEdition parameter list is negative.
C20B0002	Osi_structure size incorrect; control block length value passed in parameter list is outside the minimum/maximum range allowed. Ensure that the s_osi data structure has not changed in size in OpenEdition.
C20B0003	Insufficient memory to get SAW; sockets API work element control block cannot be built due to storage constraints.
C20B0004	Osi_wait function unavailable; invalid address for OSI services (OSIT).
C20B0005	Osi_upda function unavailable; invalid address for OSI services (OSIT).
C20B0006	Osi_upda function unavailable; invalid address for OSI services (OSIT).
C20B01xx	Osi_wait call failed; event code is SETUPSIG; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C20B02xx	Osi_wait call failed; event code is SUSPEND; xx is return code from Osi_wait. Return codes from individual calls are documented in the IBM manual <i>OpenEdition MVS File System Interface Reference</i> .
C20B03xx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B04xx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B05xx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B06xx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B07xx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.

Four-Byte Code	Description
C20B08xx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B09xx	Get Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B0Axx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B0Bxx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.
C20B0Cxx	Free Latch failed; xx is Ilatch return code. See <a href="#">Ilatch Errors</a> for more information.

## C20Cxxxx T010SWR

The following table lists X'C20C' instance codes.

Four-Byte Code	Description
C20C0001	Write length parameter is negative
C20C0002	Osi_structure is too small
C20C0003	Osi_structure is too large
C20C0004	Insufficient memory to obtain SAW control block
C20C0005	Osi_wait function not available
C20C0006	Insufficient memory to obtain secondary SAWs
C20C0007	Insufficient memory to obtain secondary SAWs
C20C0008	Insufficient memory to obtain secondary MBUFs
C20C0009	Insufficient memory to obtain secondary MBUFs
C20C000A	Insufficient memory to obtain secondary MBUFs
C20C000C	Osi_upda function not available
C20C000D	Osi_upda function not available
C20C01xx	Osi_wait SETUPSIG failed; xx is return code
C20C02xx	Osi_wait SUSPEND failed; xx is return code

Four-Byte Code	Description
C20C03xx	Get latch failed; xx is Ilatch return code
C20C04xx	Get latch failed; xx is Ilatch return code
C20C05xx	Get latch failed; xx is Ilatch return code
C20C06xx	Free latch failed; xx is Ilatch return code
C20C07xx	Free latch failed; xx is Ilatch return code
C20C08xx	Free latch failed; xx is Ilatch return code
C20C09xx	Call to T01SMOVE failed; xx is return code
C20C0Axx	Call to T01SMOVE failed; xx is return code
C20C0Bxx	Call to T01SMOVE failed; xx is return code
C20C0Cxx	Get latch failed; xx is Ilatch return code
C20C0Dxx	Get latch failed; xx is Ilatch return code
C20C0Exx	Get latch failed; xx is Ilatch return code
C20C10xx	Free latch failed; xx is Ilatch return code
C20C11xx	Free latch failed; xx is Ilatch return code
C20C12xx	Free latch failed; xx is Ilatch return code
C20C13xx	Get latch failed; xx is Ilatch return code
C20C14xx	Free latch failed; xx is Ilatch return code
C20C15xx	Get latch failed; xx is Ilatch return code
C20C16xx	Free latch failed; xx is Ilatch return code

## C20Dxxxx T010SVR

The following table lists X'C20D' instance codes.

Four-Byte Code	Description
C20D0001	I/O vector dimension is invalid
C20D0002	I/O vector array address is zero
C20D0003	I/O vector length is less than zero
C20D0004	Osi_structure is incorrect size
C20D0005	Insufficient memory to obtain SAW control block
C20D0006	Osi_wait function not available
C20D0007	I/O vector length is too large
C20D0008	Osi_upda function not available
C20D0009	Osi_upda function not available
C20D01xx	Osi_wait SETUPSIG failed; xx is return code
C20D02xx	Osi_wait SUSPEND failed; xx is return code
C20D03xx	Get latch failed; xx is Ilatch return code
C20D04xx	Get latch failed; xx is Ilatch return code
C20D05xx	Free latch failed; xx is Ilatch return code
C20D06xx	Free latch failed; xx is Ilatch return code
C20D07xx	Free latch failed; xx is Ilatch return code
C20D08xx	Get latch failed; xx is Ilatch return code
C20D09xx	Free latch failed; xx is Ilatch return code
C20D0Axx	Get latch failed; xx is Ilatch return code
C20D0Bxx	Free latch failed; xx is Ilatch return code
C20D0Cxx	Free latch failed; xx is Ilatch return code

## C20Exxxx T010SVW

The following table lists X'C20E' instance codes.

Four-Byte Code	Description
C20E0001	I/O vector dimension is not positive
C20E0002	I/O vector dimension is too large
C20E0003	I/O vector array address is zero
C20E0004	I/O vector length is less than zero
C20E0005	I/O vector length is too large
C20E0006	I/O vector length is less than zero
C20E0007	Osi_structure is too small
C20E0008	Osi_structure is too large
C20E0009	Insufficient memory to obtain SAW control block
C20E000A	Osi_wait function not available
C20E000B	Insufficient memory to obtain secondary SAWs
C20E000C	Insufficient memory to obtain secondary SAWs
C20E000D	Insufficient memory to obtain secondary MBUFs
C20E000E	Insufficient memory to obtain secondary MBUFs
C20E000F	Insufficient memory to obtain secondary MBUFs
C20E0010	Osi_upda function not available
C20E0011	Osi_upda function not available
C20E01xx	Osi_wait SETUPSIG failed; xx is return code
C20E02xx	Osi_wait SUSPEND failed; xx is return code
C20E03xx	Get latch failed; xx is Ilatch return code
C20E04xx	Get latch failed; xx is Ilatch return code
C20E05xx	Get latch failed; xx is Ilatch return code
C20E06xx	Free latch failed; xx is Ilatch return code
C20E07xx	Free latch failed; xx is Ilatch return code
C20E08xx	Free latch failed; xx is Ilatch return code
C20E09xx	Call to T01SMOVE failed; xx is return code

Four-Byte Code	Description
C20E0Axx	Call to T01SMOVE failed; xx is return code
C20E0Bxx	Call to T01SMOVE failed; xx is return code
C20E0Cxx	Get latch failed; xx is Ilatch return code
C20E0Dxx	Get latch failed; xx is Ilatch return code
C20E0Exx	Get latch failed; xx is Ilatch return code
C20E10xx	Free latch failed; xx is Ilatch return cod
C20E11xx	Free latch failed; xx is Ilatch return code
C20E12xx	Free latch failed; xx is Ilatch return code
C20E13xx	Get latch failed; xx is Ilatch return code
C20E14xx	Free latch failed; xx is Ilatch return code
C20E15xx	Get latch failed; xx is Ilatch return code
C20E16xx	Free latch failed; xx is Ilatch return code

## C20Fxxxx T010SRV

The following table lists X'C20F' instance codes.

Four-Byte Code	Description
C20F0001	Receive length parameter is negative
C20F0002	Osi_structure is incorrect size
C20F0003	Invalid flag parameter settings
C20F0004	Insufficient memory to obtain SAW control block
C20F0005	Osi_wait function not available
C20F0006	Osi_upda function not available
C20F0007	Osi_upda function not available
C20F01xx	Osi_wait SETUPSIG failed; xx is return code
C20F02xx	Osi_wait SUSPEND failed; xx is return code
C20F03xx	Get latch failed; xx is Ilatch return code

Four-Byte Code	Description
C20F04xx	Get latch failed; xx is Ilatch return code
C20F05xx	Free latch failed; xx is Ilatch return code
C20F06xx	Free latch failed; xx is Ilatch return code
C20F07xx	Free latch failed; xx is Ilatch return code
C20F08xx	Get latch failed; xx is Ilatch return code
C20F09xx	Get latch failed; xx is Ilatch return code
C20F0Axx	Free latch failed; xx is Ilatch return code
C20F0Bxx	Free latch failed; xx is Ilatch return code
C20F0Cxx	Free latch failed; xx is Ilatch return code

## C210xxxx T010SSD

The following table lists X'C210' instance codes.

Four-Byte Code	Description
C2100001	Send length parameter is negative
C2100002	Osi_structure is too small
C2100003	Osi_structure is too large
C2100004	Invalid flag parameter settings
C2100005	Insufficient memory to obtain SAW control block
C2100006	Osi_wait function not available
C2100007	Insufficient memory to obtain secondary SAWs
C2100008	Insufficient memory to obtain secondary SAWs
C2100009	Insufficient memory to obtain secondary MBUFs
C210000A	Insufficient memory to obtain secondary MBUFs
C210000B	Insufficient memory to obtain secondary MBUFs
C210000C	Osi_upda function not available
C210000D	Osi_upda function not available

Four-Byte Code	Description
C21001xx	Osi_wait SETUPSIG failed; xx is return code
C21002xx	Osi_wait SUSPEND failed; xx is return code
C21003xx	Get latch failed; xx is Ilatch return code
C21004xx	Get latch failed; xx is Ilatch return code
C21005xx	Get latch failed; xx is Ilatch return code
C21006xx	Free latch failed; xx is Ilatch return code
C21007xx	Free latch failed; xx is Ilatch return code
C21008xx	Free latch failed; xx is Ilatch return code
C21009xx	Call to T01SMOVE failed; xx is return code
C2100Axx	Call to T01SMOVE failed; xx is return code
C2100Bxx	Call to T01SMOVE failed; xx is return code
C2100Cxx	Get latch failed; xx is Ilatch return code
C2100Dxx	Get latch failed; xx is Ilatch return code
C2100Exx	Get latch failed; xx is Ilatch return code
C21010xx	Free latch failed; xx is Ilatch return code
C21011xx	Free latch failed; xx is Ilatch return code
C21012xx	Free latch failed; xx is Ilatch return code
C21013xx	Get latch failed; xx is Ilatch return code
C21014xx	Free latch failed; xx is Ilatch return code
C21015xx	Get latch failed; xx is Ilatch return code
C21016xx	Free latch failed; xx is Ilatch return code



## C211xxxx T010SFR

The following table lists X'C211' instance codes.

Four-Byte Code	Description
C2110001	Recvfrom length parameter is negative
C2110002	Osi_structure is incorrect size
C2110003	Invalid flag parameter settings
C2110004	Insufficient memory to obtain SAW control block
C2110005	Osi_wait function not available
C21101xx	Osi_wait SETUPSIG failed; xx is return code
C21102xx	Osi_wait SUSPEND failed; xx is return code
C21103xx	Get latch failed; xx is Ilatch return code
C21104xx	Get latch failed; xx is Ilatch return code
C21105xx	Free latch failed; xx is Ilatch return code
C21106xx	Free latch failed; xx is Ilatch return code
C21107xx	Free latch failed; xx is Ilatch return code
C21108xx	Osi_upda failed; xx is Osi_upda return code
C21109xx	Osi_upda failed; xx is Osi_upda return code
C2110Axx	Get latch failed; xx is Ilatch return code
C2110Bxx	Get latch failed; xx is Ilatch return code
C2110Cxx	Free latch failed; xx is Ilatch return code
C2110Dxx	Free latch failed; xx is Ilatch return code
C2110Exx	Free latch failed; xx is Ilatch return code

## C212xxxx T010STO

The following table lists X'C212' instance codes.

Four-Byte Code	Description
C2120001	Sendto length parameter is negative
C2120002	Osi_structure is too small
C2120003	Osi_structure is too large
C2120004	Invalid flag parameter settings
C2120005	Insufficient memory to obtain SAW control block
C2120006	Osi_wait function not available
C2120007	Insufficient memory to obtain secondary SAWs
C2120008	Insufficient memory to obtain secondary SAWs
C2120009	Insufficient memory to obtain secondary MBUFs
C212000A	Insufficient memory to obtain secondary MBUFs
C212000B	Insufficient memory to obtain secondary MBUFs
C212000C	Osi_upda function not available
C212000D	Osi_upda function not available
C212000F	Sockaddr parameter is missing
C2120010	Sockaddr structure too small (less than eight)
C21201xx	Osi_wait SETUPSIG failed; xx is return code
C21202xx	Osi_wait SUSPEND failed; xx is return code
C21203xx	Get latch failed; xx is Ilatch return code
C21204xx	Get latch failed; xx is Ilatch return code
C21205xx	Get latch failed; xx is Ilatch return code
C21206xx	Free latch failed; xx is Ilatch return code
C21207xx	Free latch failed; xx is Ilatch return code
C21208xx	Free latch failed; xx is Ilatch return code
C21209xx	Call to T01SMOVE failed; xx is return code
C2120Axx	Call to T01SMOVE failed; xx is return code
C2120Bxx	Call to T01SMOVE failed; xx is return code

Four-Byte Code	Description
C2120Cxx	Get latch failed; xx is latch return code
C2120Dxx	Get latch failed; xx is latch return code
C2120Exx	Get latch failed; xx is latch return code
C21210xx	Free latch failed; xx is latch return code
C21211xx	Free latch failed; xx is latch return code
C21212xx	Free latch failed; xx is latch return code
C21213xx	Get latch failed; xx is latch return code
C21214xx	Get latch failed; xx is latch return code
C21215xx	Free latch failed; xx is latch return code

## C213xxxx T010SRM

The following table lists X'C213' instance codes.

Four-Byte Code	Description
C2130001	MSG header parameter is missing
C2130002	I/O vector array address is zero
C2130003	I/O vector dimension is invalid
C2130004	Osi_structure is incorrect size
C2130005	Invalid flag parameter settings
C2130006	I/O vector length is less than zero
C2130007	Insufficient memory to obtain SAW control block
C2130008	Osi_wait function not available
C2130009	I/O vector length too large
C213000A	Osi_upda function not available
C213000B	Osi_upda function not available
C21301xx	Osi_wait SETUPSIG failed; xx is return code
C21302xx	Osi_wait SUSPEND failed; xx is return code

Four-Byte Code	Description
C21303xx	Get latch failed; xx is latch return code
C21304xx	Get latch failed; xx is latch return code
C21305xx	Free latch failed; xx is latch return code
C21306xx	Free latch failed; xx is latch return code
C21307xx	Free latch failed; xx is latch return code
C21308xx	Get latch failed; xx is latch return code
C21309xx	Free latch failed; xx is latch return code
C2130Axx	Get latch failed; xx is latch return code
C2130Bxx	Free latch failed; xx is latch return code
C2130Cxx	Free latch failed; xx is latch return code

## C214xxxx T010SSM

The following table lists X'C214' instance codes.

Four-Byte Code	Description
C2140001	MSG header parameter is missing
C2140002	I/O vector dimension is not positive
C2140003	I/O vector array address is zero
C2140004	I/O vector dimension is too large
C2140005	I/O vector length is less than zero
C2140006	I/O vector length too large
C2140007	Osi_structure is too small
C2140008	Osi_structure is too large
C2140009	Invalid flag parameter settings
C214000A	I/O vector length is less than zero
C214000B	Insufficient memory to obtain SAW control block
C214000C	Osi_wait function not available

Four-Byte Code	Description
C214000D	Insufficient memory to obtain secondary SAWs
C214000E	Insufficient memory to obtain secondary SAWs
C214000F	Insufficient memory to obtain secondary MBUFs
C2140010	Insufficient memory to obtain secondary MBUFs
C2140011	Insufficient memory to obtain secondary MBUFs
C2140012	Osi_upda function not available
C2140013	Osi_upda function not available
C2140015	MSG header parameter is too small
C2140016	MSG name address is missing
C2140017	MSG name length is too small (less than eight)
C21401xx	Osi_wait SETUPSIG failed; xx is return code
C21402xx	Osi_wait SUSPEND failed; xx is return code
C21403xx	Get latch failed; xx is latch return code
C21404xx	Get latch failed; xx is latch return code
C21405xx	Get latch failed; xx is latch return code
C21406xx	Free latch failed; xx is latch return code
C21407xx	Free latch failed; xx is latch return code
C21408xx	Free latch failed; xx is latch return code
C21409xx	Call to T01SMOVE failed; xx is return code
C2140Axx	Call to T01SMOVE failed; xx is return code
C2140Bxx	Call to T01SMOVE failed; xx is return code
C2140Cxx	Get latch failed; xx is latch return code
C2140Dxx	Get latch failed; xx is latch return code
C2140Exx	Get latch failed; xx is latch return code
C21410xx	Free latch failed; xx is latch return code
C21411xx	Free latch failed; xx is latch return code
C21412xx	Free latch failed; xx is latch return code
C21413xx	Get latch failed; xx is latch return code
C21414xx	Free latch failed; xx is latch return code

Four-Byte Code	Description
C21415xx	Get latch failed; xx is latch return code
C21416xx	Free latch failed; xx is latch return code

## C215xxx T010SSL

The following table lists X'C215' instance codes.

Four-Byte Code	Description
C2150001	Invalid OE function; neither select nor batsel
C2150002	File count is zero for batsel
C2150003	Insufficient memory to obtain SAW control block
C2150004	Invalid batsel; neither query nor cancel
C2150005	Invalid select; neither query nor cancel
C2150006	Insufficient memory to obtain SAW control block
C2150007	Invalid SEPM control block
C2150008	TCP was restarted prior to this request
C21501xx	Get latch failed; xx is latch return code
C21502xx	Get latch failed; xx is latch return code
C21503xx	Free latch failed; xx is latch return code
C21504xx	Free latch failed; xx is latch return code
C21505xx	Free latch failed; xx is latch return code

## C216xxxx T010SSN

The following table lists X'C216' instance codes.

Four-Byte Code	Description
C2160001	Invalid function socket type (Stream)
C2160003	Insufficient memory to obtain SAW control block
C2160004	Sockaddr structure too small (less than eight)
C21601xx	Get latch failed; xx is llatch return code
C21602xx	Free latch failed; xx is llatch return code

## C217xxxx T010SSH

The following table lists X'C217' instance codes.

Four-Byte Code	Description
C2170001	Insufficient memory to obtain SAW control block
C21701xx	Get latch failed; xx is llatch return code
C21702xx	Free latch failed; xx is llatch return code

## C218xxxx T010SGH

The following table lists X'C218' instance codes.

Four-Byte Code	Description
C2180001	Gethost failed; NETB control block not found
C2180002	Gethost failed; Name parameter address is zero

## C219xxxx T010SCA

The following table lists X'C219' instance codes.

Four-Byte Code	Description
C2190001	Async I/O function was cancelled
C2190002	Async I/O function was not found
C21901xx	Get latch failed; xx is latch return code
C21902xx	Free latch failed; xx is latch return code
C21903xx	Free latch failed; xx is latch return code

## C220xxxx T011SPC

The following table lists X'C220' instance codes.

Four-Byte Code	Description
C2200001	Socket function out of range
C2200002	Unsupported socket function
C2200003	No Index1
C2200004	No Index2
C2200005	No IPTH
C2200007	Socket number out of range
C2200008	No IPTHSPTR
C2200009	No Index4
C220000A	No SEPM
C220000B	Socket taken but not Close()
C220000C	Socket passed but not Close()
C220000D	End point terminating
C220000E	Multiple requests for APITYPE=2
C220000F	Caller specified IPNORPY



C22000FD	Severing connection attempt
C22000FE	Severing a path
C22000FF	Severing - no IPTH to reply on

## C221xxx T011SAC

The following table lists X'C221' instance codes.

Four-Byte Code	Description
C2210000	Return pending callback
C2210001	IUCV is down
C2210002	Unable to obtain SAW
C2210003	IUCV header error
C2210004	Socket number out of range
C2210005	No Index4
C2210006	Socket in use
C2210007	SAVXLTCF FREE error. XACPT says suspend
C2210008	SEPMLTCF FREE error. XACPT says suspend
C2210009	Accept request exceeds maximum sockets allowed.
C221000A	Accept refused, Unicenter TCPaccess is shutting down.
C2210106	IUCV error - protection exception
C2210107	IUCV error - addressing exception
C22101nn	IUCV error - IUCV IPRCODE nn
C2210201	T02CIUCV RC = 01. IPRCODE set
C2210203	T02CIUCV RC = 03. Not using IUCV services
C2210204	T02CIUCV RC = 04. Protection exception
C2210205	T02CIUCV RC = 05. Addressing exception
C2210206	T02CIUCV RC = 06. Specification exception
C2210207	T02CIUCV RC = 07. Operation exception
C2210208	T02CIUCV RC = 08. Invalid function code

Four-Byte Code	Description
C2210209	T02CIUCV RC = 09. IUCV terminating
C221020A	T02CIUCV RC = 10. FRR active on entry
C22103nn	GET SAVXLTCH error code base code
C2210304	Latch held exclusively
C2210308	Logic error
C221030C	Bad parameter
C2210310	User holding the CML
C2210314	Suspend failed
C2210318	Storage obtain failure
C221031C	SFRE entry in use
C2210320	Latch held shared, requested exclusive
C2210324	Latch held exclusively, requested shared
C22104nn	GET SEPMLTCH error base code
C2210404	Latch held exclusively
C2210408	Logic error
C221040C	Bad parameter
C2210410	User holding the CML
C2210414	Suspend failed
C2210418	Storage obtain failure
C221041C	SFRE entry in use
C2210420	Latch held shared, requested exclusive
C2210424	Latch held exclusively, requested shared
C22105nn	FREE SAVXLTCH base error code
C2210504	Latch not held
C2210508	Logic error
C221050C	Bad parameter
C22106nn	FREE SEPMLTCH base error code
C2210604	Latch not held
C2210608	Logic error

Four-Byte Code	Description
C221060C	Bad parameter
C22107nn	GET SEPMLTCH error base code (parent SEPM)
C2210704	Latch held exclusively
C2210708	Logic error
C221070C	Bad parameter
C2210710	User holding the CML
C2210714	Suspend failed
C2210718	Storage obtain failure
C221071C	SFRE entry in use
C2210720	Latch held shared, requested exclusive
C2210724	Latch held exclusively, requested shared
C22108nn	FREE SEPMLTCH base error code (child SEPM)
C2210804	Latch not held
C2210808	Logic error
C221080C	Bad parameter
C2210999	IUCV purged message
C2210Ann	GET SAVXLTCH error base code (parent SEPM)
C2210A04	Latch held exclusively
C2210A08	Logic error
C2210A0C	Bad parameter
C2210A10	User holding the CML
C2210A14	Suspend failed
C2210A18	Storage obtain failure
C2210A1C	SFRE entry in use
C2210A20	Latch held shared, requested exclusive
C2210A24	Latch held exclusively, requested shared
C2210Bnn	FREE SAVXLTCH base error code (child SEPM)
C2210B04	Latch not held

Four-Byte Code	Description
C2210B08	Logic error
C2210B0C	Bad parameter

## C222xxxx T011SBD

The following table lists X'C222' instance codes.

Four-Byte Code	Description
C2220002	Answer area too small
C2220015	Sockaddr length is too short
C22200E1	Unable to obtain SAW
C22200FF	IUCV terminated
C22201nn	IUCV receive error base code
C2220101	Invalid path ID
C2220102	Path quiesced
C2220103	Message limit exceeded
C2220104	Priority message not allowed on this path
C2220105	Buffer too short for message
C2220106	Fetch protection exception
C2220107	Addressing exception
C2220108	Class or path invalid
C2220109	Message has been purged
C222010A	Message length negative
C222010B	Target is not in system
C222010C	Target is not prepared for IUCV
C222010D	Invoker max connect count exceeded
C222010E	Target max connect count exceeded
C222010F	Not authorized to communicate with target

Four-Byte Code	Description
C2220110	Invalid system service name
C2220111	Invalid function code
C2220112	Invalid message limit
C2220113	Duplicate buffer declaration
C2220114	Path has been severed
C2220115	Parm list message not allowed
C2220116	Send list invalid
C2220117	Negative length in list
C2220118	Invalid total list length
C2220119	PRMMSG and BUF/ANSLIST not allowed
C222011A	Buffer list not double-word aligned
C222011B	Answer list not double-word aligned
C222011C	No control buffer exists
C2220130	Function not supported for CSS
C22202 $nn$	IUCV reply error base code
C2220201	Invalid path ID
C2220202	Path quiesced
C2220203	Message limit exceeded
C2220204	Priority message not allowed on this path
C2220205	Buffer too short for message
C2220206	Fetch protection exception
C2220207	Addressing exception
C2220208	Class or path invalid
C2220209	Message has been purged
C222020A	Message length negative
C222020B	Target is not in system
C222020C	Target is not prepared for IUCV
C222020D	Invoker max connect count exceeded
C222020E	Target max connect count exceeded

Four-Byte Code	Description
C222020F	Not authorized to communicate with target
C2220210	Invalid system service name
C2220211	Invalid function code
C2220212	Invalid message limit
C2220213	Duplicate buffer declaration
C2220214	Path has been severed
C2220215	Parm list message not allowed
C2220216	Send list invalid
C2220217	Negative length in list
C2220218	Invalid total list length
C2220219	PRMMMSG and BUF/ANSLIST not allowed
C222021A	Buffer list not double-word aligned
C222021B	Answer list not double-word aligned
C222021C	No control buffer exists
C2220230	Function not supported for CSS
C22203nn	Get Latch error base code
C2220304	Latch held exclusive, request was exclusive
C2220308	Logic error
C222030C	Bad parameter
C2220310	User held the CML
C2220314	Suspend failed
C2220318	Storage obtain failed
C222031C	SFRE entry in use
C2220320	Latch held shared, request was exclusive
C2220324	Latch held exclusive, request was shared
C22204nn	Free Latch error base code
C2220404	Latch not held
C2220408	Logic error
C222040C	Bad parameter

## C223xxxx T011SCA

The following table lists X'C223' instance codes.

Four-Byte Code	Description
C2230001	SAW not found
C2230004	Cannot obtain primary SAW
C2230005	APItype not three for cancel
C2230006	Message was not inline data
C2230008	Invalid Path ID specified
C2230009	Reply area is too small
C2230014	Invalid call to be cancelled
C223000A	IUCV callback failed
C223000B	No matching socket
C223000C	No SAW on receive queue
C2230020	SAW queue in use
C2230021	Empty SAW chain
C22300F0	IUCV not available
C22302nn	Get SEPMLTCH failure while canceling
C2230204	Latch held exclusive, request was exclusive
C2230208	Logic error
C223020C	Bad parameter
C2230210	User held the CML
C2230214	Suspend failed
C2230218	Storage obtain failed
C223021C	SFRE entry in use
C2230218	Storage obtain failed
C2230220	Latch held shared, request was exclusive
C2230224	Latch held exclusive, request was shared
C22303nn	Get SPCGLTCH base code
C2230304	Latch held exclusive, request was exclusive

Four-Byte Code	Description
C2230308	Logic error
C223030C	Bad parameter
C2230310	User held the CML
C2230314	Suspend failed
C2230318	Storage obtain failed
C223031C	SFRE entry in use
C2230318	Storage obtain failed
C2230320	Latch held shared, request was exclusive
C2230324	Latch held exclusive, request was shared
C22304nn	Get SEPMLTCH failure
C2230404	Latch held exclusive, request was exclusive
C2230408	Logic error
C223040C	Bad parameter
C2230410	User held the CML
C2230414	Suspend failed
C2230418	Storage obtain failed
C223041C	SFRE entry in use
C2230418	Storage obtain failed
C2230420	Latch held shared, request was exclusive
C2230424	Latch held exclusive, request was shared
C22305nn	Free SEPMLTCH failure base code
C2230504	Latch not held
C2230508	Logic error
C223050C	Bad parameter
C22306nn	Free SPCBLTCH failure base code after queue scan
C2230604	Latch not held
C2230608	Logic error
C223060C	Bad parameter
C22307nn	Free SPCBLTCH failure base code



<b>Four-Byte Code</b>	<b>Description</b>
C2230704	Latch not held
C2230708	Logic error
C223070C	Bad parameter
C2230D00	IUCV reply failed
C2230Enn	IUCV reply error base code
C2230E01	Invalid path ID
C2230E02	Path quiesced
C2230E03	Message limit exceeded
C2230E04	Priority message not allowed on this path
C2230E05	Buffer too short for message
C2230E06	Fetch protection exception
C2230E07	Addressing exception
C2230E08	Class or path invalid
C2230E09	Message has been purged
C2230E0A	Message length negative
C2230E0B	Target is not in system
C2230E0C	Target is not prepared for IUCV
C2230E0D	Invoker max connect count exceeded
C2230E0E	Target max connect count exceeded
C2230E0F	Not authorized to communicate with target
C2230E10	Invalid system service name
C2230E11	Invalid function code
C2230E12	Invalid message limit
C2230E13	Duplicate buffer declaration
C2230E14	Path has been severed
C2230E15	Parm list message not allowed
C2230E16	Send list invalid
C2230E17	Negative length in list
C2230E18	Invalid total list length

Four-Byte Code	Description
C2230E19	PRMMMSG and BUF/ANSLIST not allowed
C2230E1A	Buffer list not double-word aligned
C2230E1B	Answer list not double-word aligned
C2230E1C	No control buffer exists
C2230E30	Function not supported for CSS

## C224xxxx T011SCL

The following table lists X'C224' instance codes.

Four-Byte Code	Description
C2240000	Return to caller pending callback
C2240001	IUCV is not available
C2240002	Unable to obtain SAW
C2240003	IUCV header error
C22401nn	IUCV error base code if IPRCODE was saved
C2240101	Invalid path ID
C2240102	Path quiesced
C2240103	Message limit exceeded
C2240104	Priority message not allowed on this path
C2240105	Buffer too short for message
C2240106	Fetch protection exception
C2240107	Addressing exception
C2240108	Class or path invalid
C2240109	Message has been purged
C224010A	Message length negative
C224010B	Target is not in system
C224010C	Target is not prepared for IUCV

Four-Byte Code	Description
C224010D	Invoker max connect count exceeded
C224010E	Target max connect count exceeded
C224010F	Not authorized to communicate with target
C2240110	Invalid system service name
C2240111	Invalid function code
C2240112	Invalid message limit
C2240113	Duplicate buffer declaration
C2240114	Path has been severed
C2240115	Parm list message not allowed
C2240116	Send list invalid
C2240117	Negative length in list
C2240118	Invalid total list length
C2240119	PRMMSG and BUF/ANSLIST not allowed
C224011A	Buffer list not double-word aligned
C224011B	Answer list not double-word aligned
C224011C	No control buffer exists
C2240130	Function not supported for CSS
C22402nn	IUCV error base code if IPRCODE was not saved
C2240201	Invalid path ID
C2240202	Path quiesced
C2240203	Message limit exceeded
C2240204	Priority message not allowed on this path
C2240205	Buffer too short for message
C2240206	Fetch protection exception
C2240207	Addressing exception
C2240208	Class or path invalid
C2240209	Message has been purged
C224020A	Message length negative
C224020B	Target is not in system

Four-Byte Code	Description
C224020C	Target is not prepared for IUCV
C224020D	Invoker max connect count exceeded
C224020E	Target max connect count exceeded
C224020F	Not authorized to communicate with target
C2240210	Invalid system service name
C2240211	Invalid function code
C2240212	Invalid message limit
C2240213	Duplicate buffer declaration
C2240214	Path has been severed
C2240215	Parm list message not allowed
C2240216	Send list invalid
C2240217	Negative length in list
C2240218	Invalid total list length
C2240219	PRMMMSG and BUF/ANSLIST not allowed
C224021A	Buffer list not double-word aligned
C224021B	Answer list not double-word aligned
C224021C	No control buffer exists
C2240230	Function not supported for CSS
C22403nn	Get SEPMLTCH failure base code
C2240304	Latch held exclusive, request was exclusive
C2240308	Logic error
C224030C	Bad parameter
C2240310	User held the CML
C2240314	Suspend failed
C2240318	Storage obtain failed
C224031C	SFRE entry in use
C2240318	Storage obtain failed
C2240320	Latch held shared, request was exclusive
C2240324	Latch held exclusive, request was shared

Four-Byte Code	Description
C22405nn	Free SEPMLTCH failure base code
C2240504	Latch not held
C2240508	Logic error
C224050C	Bad parameter
C22406nn	Free SEPMLTCH failure base code, while processing returned SAWs
C2240604	Latch not held
C2240608	Logic error
C224060C	Bad parameter
C2240699	IUCV message was purged

## C225xxxx T011SCN

The following table lists X'C225' instance codes.

Four-Byte Code	Description
C2250000	Return to caller pending callback
C2250001	IUCV header error
C2250002	Answer area too small
C2250015	Sockaddr data too short
C22500E1	Failure to obtain SAW
C22500FF	IUCV is not available
C22501nn	IUCV receive error base code
C2250101	Invalid path ID
C2250102	Path quiesced
C2250103	Message limit exceeded
C2250104	Priority message not allowed on this path
C2250105	Buffer too short for message
C2250106	Fetch protection exception

Four-Byte Code	Description
C2250107	Addressing exception
C2250108	Class or path invalid
C2250109	Message has been purged
C225010A	Message length negative
C225010B	Target is not in system
C225010C	Target is not prepared for IUCV
C225010D	Invoker max connect count exceeded
C225010E	Target max connect count exceeded
C225010F	Not authorized to communicate with target
C2250110	Invalid system service name
C2250111	Invalid function code
C2250112	Invalid message limit
C2250113	Duplicate buffer declaration
C2250114	Path has been severed
C2250115	Parm list message not allowed
C2250116	Send list invalid
C2250117	Negative length in list
C2250118	Invalid total list length
C2250119	PRMMSG and BUF/ANSLIST not allowed
C225011A	Buffer list not double-word aligned
C225011B	Answer list not double-word aligned
C225011C	No control buffer exists
C2250130	Function not supported for CSS
C22502nn	IUCV reply error base code
C2250201	Invalid path ID
C2250202	Path quiesced
C2250203	Message limit exceeded
C2250204	Priority message not allowed on this path
C2250205	Buffer too short for message

Four-Byte Code	Description
C2250206	Fetch protection exception
C2250207	Addressing exception
C2250208	Class or path invalid
C2250209	Message has been purged
C225020A	Message length negative
C225020B	Target is not in system
C225020C	Target is not prepared for IUCV
C225020D	Invoker max connect count exceeded
C225020E	Target max connect count exceeded
C225020F	Not authorized to communicate with target
C2250210	Invalid system service name
C2250211	Invalid function code
C2250212	Invalid message limit
C2250213	Duplicate buffer declaration
C2250214	Path has been severed
C2250215	Parm list message not allowed
C2250216	Send list invalid
C2250217	Negative length in list
C2250218	Invalid total list length
C2250219	PRMMSG and BUF/ANSLIST not allowed
C225021A	Buffer list not double-word aligned
C225021B	Answer list not double-word aligned
C225021C	No control buffer exists
C2250230	Function not supported for CSS
C22503nn	Get Latch error base code
C2250304	Latch held exclusive, request was exclusive
C2250308	Logic error
C225030C	Bad parameter
C2250310	User held the CML

Four-Byte Code	Description
C2250314	Suspend failed
C2250318	Storage obtain failed
C225031C	SFRE entry in use
C2250318	Storage obtain failed
C2250320	Latch held shared, request was exclusive
C2250324	Latch held exclusive, request was shared
C22504nn	Free Latch error base code
C2250404	Latch not held
C2250408	Logic error
C225040C	Bad parameter

## C226xxxx T011SCP

The following table lists X'C226' instance codes.

Four-Byte Code	Description
C226000A	EIB type is not connect pending
C226000B	Option MSGDATA not set
C226000C	Option quiesce was set
C226000D	Option priority-msg was set
C226000E	User data was not zeros
C226000F	Path ID flag was not set
C2260020	Unable to obtain IPATH block
C2260030	Unable to obtain index block
C2260050	Invalid Path ID
C2260E00	Terminating
C2260F00	IUCV not available
C22620nn	IUCV accept error base code



<b>Four-Byte Code</b>	<b>Description</b>
C2262001	Invalid path ID
C2262002	Path quiesced
C2262003	Message limit exceeded
C2262004	Priority message not allowed on this path
C2262005	Buffer too short for message
C2262006	Fetch protection exception
C2262007	Addressing exception
C2262008	Class or path invalid
C2262009	Message has been purged
C226200A	Message length negative
C226200B	Target is not in system
C226200C	Target is not prepared for IUCV
C226200D	Invoker max connect count exceeded
C226200E	Target max connect count exceeded
C226200F	Not authorized to communicate with target
C2262010	Invalid system service name
C2262011	Invalid function code
C2262012	Invalid message limit
C2262013	Duplicate buffer declaration
C2262014	Path has been severed
C2262015	Parm list message not allowed
C2262016	Send list invalid
C2262017	Negative length in list
C2262018	Invalid total list length
C2262019	PRMMSG and BUF/ANSLIST not allowed
C226201A	Buffer list not double-word aligned
C226201B	Answer list not double-word aligned
C226201C	No control buffer exists
C2262030	Function not supported for CSS

Four-Byte Code	Description
C22630mm	IUCV sever error base code
C2263001	Invalid path ID
C2263002	Path quiesced
C2263003	Message limit exceeded
C2263004	Priority message not allowed on this path
C2263005	Buffer too short for message
C2263006	Fetch protection exception
C2263007	Addressing exception
C2263008	Class or path invalid
C2263009	Message has been purged
C226300A	Message length negative
C226300B	Target is not in system
C226300C	Target is not prepared for IUCV
C226300D	Invoker max connect count exceeded
C226300E	Target max connect count exceeded
C226300F	Not authorized to communicate with target
C2263010	Invalid system service name
C2263011	Invalid function code
C2263012	Invalid message limit
C2263013	Duplicate buffer declaration
C2263014	Path has been severed
C2263015	Parm list message not allowed
C2263016	Send list invalid
C2263017	Negative length in list
C2263018	Invalid total list length
C2263019	PRMMSG and BUF/ANSLIST not allowed

Four-Byte Code	Description
C226301A	Buffer list not double-word aligned
C226301B	Answer list not double-word aligned
C226301C	No control buffer exists
C2263030	Function not supported for CSS

## C227xxx T011SGH

The following table lists X'C227' instance codes.

Four-Byte Code	Description
C2270001	IUCV not available
C2270002	Unable to obtain SAW
C2270003	IUCV header error
C22701nn	IUCV error base code if IPRCODE was saved
C2270101	Invalid path ID
C2270102	Path quiesced
C2270103	Message limit exceeded
C2270104	Priority message not allowed on this path
C2270105	Buffer too short for message
C2270106	Fetch protection exception
C2270107	Addressing exception
C2270108	Class or path invalid
C2270109	Message has been purged
C227010A	Message length negative
C227010B	Target is not in system
C227010C	Target is not prepared for IUCV
C227010D	Invoker max connect count exceeded
C227010E	Target max connect count exceeded

Four-Byte Code	Description
C227010F	Not authorized to communicate with target
C2270110	Invalid system service name
C2270111	Invalid function code
C2270112	Invalid message limit
C2270113	Duplicate buffer declaration
C2270114	Path has been severed
C2270115	Parm list message not allowed
C2270116	Send list invalid
C2270117	Negative length in list
C2270118	Invalid total list length
C2270119	PRMMSG and BUF/ ANSLIST not allowed
C227011A	Buffer list not double-word aligned
C227011B	Answer list not double-word aligned
C227011C	No control buffer exists
C2270130	Function not supported for CSS
C22702nn	IUCV error base code if IPRCODE was not saved
C2270201	Invalid path ID
C2270202	Path quiesced
C2270203	Message limit exceeded
C2270204	Priority message not allowed on this path
C2270205	Buffer too short for message
C2270206	Fetch protection exception
C2270207	Addressing exception
C2270208	Class or path invalid
C2270209	Message has been purged
C227020A	Message length negative
C227020B	Target is not in system
C227020C	Target is not prepared for IUCV
C227020D	Invoker max connect count exceeded

<b>Four-Byte Code</b>	<b>Description</b>
C227020E	Target max connect count exceeded
C227020F	Not authorized to communicate with target
C2270210	Invalid system service name
C2270211	Invalid function code
C2270212	Invalid message limit
C2270213	Duplicate buffer declaration
C2270214	Path has been severed
C2270215	Parm list message not allowed
C2270216	Send list invalid
C2270217	Negative length in list
C2270218	Invalid total list length
C2270219	PRMMSG and BUF/ANSLIST not allowed
C227021A	Buffer list not double-word aligned
C227021B	Answer list not double-word aligned
C227021C	No control buffer exists
C2270230	Function not supported for CSS
C2270999	IUCV message purged

## C228xxxx T011SGN

The following table lists X'C228' instance codes.

Four-Byte Code	Description
C2280001	IUCV is not available
C2280002	Unable to obtain SAW
C2280003	IUCV header error
C22801nn	IUCV error base code if IPRCODE was saved
C2280101	Invalid path ID
C2280102	Path quiesced
C2280103	Message limit exceeded
C2280104	Priority message not allowed on this path
C2280105	Buffer too short for message
C2280106	Fetch protection exception
C2280107	Addressing exception
C2280108	Class or path invalid
C2280109	Message has been purged
C228010A	Message length negative
C228010B	Target is not in system
C228010C	Target is not prepared for IUCV
C228010D	Invoker max connect count exceeded
C228010E	Target max connect count exceeded
C228010F	Not authorized to communicate with target
C2280110	Invalid system service name
C2280111	Invalid function code
C2280112	Invalid message limit
C2280113	Duplicate buffer declaration
C2280114	Path has been severed
C2280115	Parm list message not allowed
C2280116	Send list invalid
C2280117	Negative length in list

Four-Byte Code	Description
C2280118	Invalid total list length
C2280119	PRMMSG and BUF/ ANSLIST not allowed
C228011A	Buffer list not double-word aligned
C228011B	Answer list not double-word aligned
C228011C	No control buffer exists
C2280130	Function not supported for CSS
C22802nn	IUCV error base code if IPRCODE was not saved
C2280201	Invalid path ID
C2280202	Path quiesced
C2280203	Message limit exceeded
C2280204	Priority message not allowed on this path
C2280205	Buffer too short for message
C2280206	Fetch protection exception
C2280207	Addressing exception
C2280208	Class or path invalid
C2280209	Message has been purged
C228020A	Message length negative
C228020B	Target is not in system
C228020C	Target is not prepared for IUCV
C228020D	Invoker max connect count exceeded
C228020E	Target max connect count exceeded
C228020F	Not authorized to communicate with target
C2280210	Invalid system service name
C2280211	Invalid function code
C2280212	Invalid message limit
C2280213	Duplicate buffer declaration
C2280214	Path has been severed
C2280215	Parm list message not allowed
C2280216	Send list invalid
C2280217	Negative length in list

Four-Byte Code	Description
C2280218	Invalid total list length
C2280219	PRMMSG and BUF/ ANSLIST not allowed
C228021A	Buffer list not double-word aligned
C228021B	Answer list not double-word aligned
C228021C	No control buffer exists
C2280230	Function not supported for CSS
C22803nn	Get Latch error base code
C2280304	Latch held exclusive, request was exclusive
C2280308	Logic error
C228030C	Bad parameter
C2280310	User held the CML
C2280314	Suspend failed
C2280318	Storage obtain failed
C228031C	SFRE entry in use
C2280318	Storage obtain failed
C2280320	Latch held shared, request was exclusive
C2280324	Latch held exclusive, request was shared
C22804nn	Free Latch error base code
C2280404	Latch not held
C2280408	Logic error
C228040C	Bad parameter
C2280999	IUCV message purged



## C229xxxx T011SGO

The following table lists X'C229' instance codes.

Four-Byte Code	Description
C22900E1	Unable to obtain SAW
C2290001	IUCV header error
C2290002	Answer area too small
C2290010	Incorrect format for IBM options
C22900FF	IUCV not available
C22902nn	IUCV error base code if IPRCODE was not saved
C2290201	Invalid path ID
C2290202	Path quiesced
C2290203	Message limit exceeded
C2290204	Priority message not allowed on this path
C2290205	Buffer too short for message
C2290206	Fetch protection exception
C2290207	Addressing exception
C2290208	Class or path invalid
C2290209	Message has been purged
C229020A	Message length negative
C229020B	Target is not in system
C229020C	Target is not prepared for IUCV
C229020D	Invoker max connect count exceeded
C229020E	Target max connect count exceeded
C229020F	Not authorized to communicate with target
C2290210	Invalid system service name
C2290211	Invalid function code
C2290212	Invalid message limit
C2290213	Duplicate buffer declaration
C2290214	Path has been severed

Four-Byte Code	Description
C2290215	Parm list message not allowed
C2290216	Send list invalid
C2290217	Negative length in list
C2290218	Invalid total list length
C2290219	PRMMSG and BUF/ANSLIST not allowed
C229021A	Buffer list not double-word aligned
C229021B	Answer list not double-word aligned
C229021C	No control buffer exists
C2290230	Function not supported for CSS
C22903nn	Get Latch error base code
C2290304	Latch held exclusive, request was exclusive
C2290308	Logic error
C229030C	Bad parameter
C2290310	User held the CML
C2290314	Suspend failed
C2290318	Storage obtain failed
C229031C	SFRE entry in use
C2290318	Storage obtain failed
C2290320	Latch held shared, request was exclusive
C2290324	Latch held exclusive, request was shared
C22904nn	Free Latch error base code
C2290404	Latch not held
C2290408	Logic error
C229040C	Bad parameter

## C22Axxxx T011SGS

The following table lists X'C22A' instance codes.

Four-Byte Code	Description
C22A0001	SAW found on pending queue
C22A0002	UICV is not available
C22A0003	Unable to obtain SAW
C22A0004	IUCV flags error
C22A0005	IUCV header error
C22A0006	Invalid socket type
C22A0007	Socket in listening state
C22A0008	Socket is not connected
C22A01nn	IUCV error base code if IPRCODE was saved
C22A0101	Invalid path ID
C22A0102	Path quiesced
C22A0103	Message limit exceeded
C22A0104	Priority message not allowed on this path
C22A0105	Buffer too short for message
C22A0106	Fetch protection exception
C22A0107	Addressing exception
C22A0108	Class or path invalid
C22A0109	Message has been purged
C22A010A	Message length negative
C22A010B	Target is not in system
C22A010C	Target is not prepared for IUCV
C22A010D	Invoker max connect count exceeded
C22A010E	Target max connect count exceeded
C22A010F	Not authorized to communicate with target
C22A0110	Invalid system service name
C22A0111	Invalid function code

Four-Byte Code	Description
C22A0112	Invalid message limit
C22A0113	Duplicate buffer declaration
C22A0114	Path has been severed
C22A0115	Parm list message not allowed
C22A0116	Send list invalid
C22A0117	Negative length in list
C22A0118	Invalid total list length
C22A0119	PRMMSG and BUF/ANSLIST not allowed
C22A011A	Buffer list not double-word aligned
C22A011B	Answer list not double-word aligned
C22A011C	No control buffer exists
C22A0130	Function not supported for CSS
C22A02 $nn$	IUCV error base code if IPRCODE was not saved
C22A0201	Invalid path ID
C22A0202	Path quiesced
C22A0203	Message limit exceeded
C22A0204	Priority message not allowed on this path
C22A0205	Buffer too short for message
C22A0206	Fetch protection exception
C22A0207	Addressing exception
C22A0208	Class or path invalid
C22A0209	Message has been purged
C22A020A	Message length negative
C22A020B	Target is not in system
C22A020C	Target is not prepared for IUCV
C22A020D	Invoker max connect count exceeded
C22A020E	Target max connect count exceeded
C22A020F	Not authorized to communicate with target
C22A0210	Invalid system service name

<b>Four-Byte Code</b>	<b>Description</b>
C22A0211	Invalid function code
C22A0212	Invalid message limit
C22A0213	Duplicate buffer declaration
C22A0214	Path has been severed
C22A0215	Parm list message not allowed
C22A0216	Send list invalid
C22A0217	Negative length in list
C22A0218	Invalid total list length
C22A0219	PRMMSG and BUF/ANSLIST not allowed
C22A021A	Buffer list not double-word aligned
C22A021B	Answer list not double-word aligned
C22A021C	No control buffer exists
C22A0230	Function not supported for CSS
C22A03nn	Get Latch error base code
C22A0304	Latch held exclusive, request was exclusive
C22A0308	Logic error
C22A030C	Bad parameter
C22A0310	User held the CML
C22A0314	Suspend failed
C22A0318	Storage obtain failed
C22A031C	SFRE entry in use
C22A0318	Storage obtain failed
C22A0320	Latch held shared, request was exclusive
C22A0324	Latch held exclusive, request was shared
C22A05nn	Free Latch error base code
C22A0504	Latch not held
C22A0508	Logic error
C22A050C	Bad parameter
C22A06nn	Free Latch error during error processing base code

Four-Byte Code	Description
C22A0604	Latch not held
C22A0608	Logic error
C22A060C	Bad parameter

## C22Bxxxx T011SID

The following table lists X'C22B' instance codes.

Four-Byte Code	Description
C22B0001	IUCV not available
C22B0002	Unable to obtain SAW
C22B0003	IUCV header error
C22B01nn	IUCV error base code if IPRCODE was saved
C22B0101	Invalid path ID
C22B0102	Path quiesced
C22B0103	Message limit exceeded
C22B0104	Priority message not allowed on this path
C22B0105	Buffer too short for message
C22B0106	Fetch protection exception
C22B0107	Addressing exception
C22B0108	Class or path invalid
C22B0109	Message has been purged
C22B010A	Message length negative
C22B010B	Target is not in system
C22B010C	Target is not prepared for IUCV
C22B010D	Invoker max connect count exceeded
C22B010E	Target max connect count exceeded
C22B010F	Not authorized to communicate with target

Four-Byte Code	Description
C22B0110	Invalid system service name
C22B0111	Invalid function code
C22B0112	Invalid message limit
C22B0113	Duplicate buffer declaration
C22B0114	Path has been severed
C22B0115	Parm list message not allowed
C22B0116	Send list invalid
C22B0117	Negative length in list
C22B0118	Invalid total list length
C22B0119	PRMMSG and BUF/ANSLIST not allowed
C22B011A	Buffer list not double-word aligned
C22B011B	Answer list not double-word aligned
C22B011C	No control buffer exists
C22B0130	Function not supported for CSS
C22B02nn	IUCV error base code if IPRCODE was not saved
C22B0201	Invalid path ID
C22B0202	Path quiesced
C22B0203	Message limit exceeded
C22B0204	Priority message not allowed on this path
C22B0205	Buffer too short for message
C22B0206	Fetch protection exception
C22B0207	Addressing exception
C22B0208	Class or path invalid
C22B0209	Message has been purged
C22B020A	Message length negative
C22B020B	Target is not in system
C22B020C	Target is not prepared for IUCV
C22B020D	Invoker max connect count exceeded
C22B020E	Target max connect count exceeded

<b>Four-Byte Code</b>	<b>Description</b>
C22B020F	Not authorized to communicate with target
C22B0210	Invalid system service name
C22B0211	Invalid function code
C22B0212	Invalid message limit
C22B0213	Duplicate buffer declaration
C22B0214	Path has been severed
C22B0215	Parm list message not allowed
C22B0216	Send list invalid
C22B0217	Negative length in list
C22B0218	Invalid total list length
C22B0219	PRMMMSG and BUF/ ANSLIST not allowed
C22B021A	Buffer list not double-word aligned
C22B021B	Answer list not double-word aligned
C22B021C	No control buffer exists
C22B0230	Function not supported for CSS
C22B0999	IUCV message purged



## C22Cxxxx T011SIN

The following table lists X'C22C' instance codes.

Four-Byte Code	Description
C22C0009	Second initial message call
C22C000A	EIB TRGCLS not zero initially
C22C000B	Option MSGDATA was set
C22C000C	Option one-way was set
C22C000D	Path ID configuration error
C22C0020	Initial message length invalid
C22C0021	Initial replay length invalid
C22C0023	API name is not IUCVAPI
C22C0024	API type is not 2 or 3
C22C0025	Task ID is blank
C22C0028	Duplicate Jobname/TaskID
C22C0030	Unable to obtain Index3
C22C0031	Unable to obtain Index4
C22C0E00	Terminating
C22C0F00	IUCV not available
C22C10nn	IUCV receive error base code
C22C1001	Invalid path ID
C22C1002	Path quiesced
C22C1003	Message limit exceeded
C22C1004	Priority message not allowed on this path
C22C1005	Buffer too short for message
C22C1006	Fetch protection exception
C22C1007	Addressing exception
C22C1008	Class or path invalid
C22C1009	Message has been purged
C22C100A	Message length negative

Four-Byte Code	Description
C22C100B	Target is not in system
C22C100C	Target is not prepared for IUCV
C22C100D	Invoker max connect count exceeded
C22C100E	Target max connect count exceeded
C22C100F	Not authorized to communicate with target
C22C1010	Invalid system service name
C22C1011	Invalid function code
C22C1012	Invalid message limit
C22C1013	Duplicate buffer declaration
C22C1014	Path has been severed
C22C1015	Parm list message not allowed
C22C1016	Send list invalid
C22C1017	Negative length in list
C22C1018	Invalid total list length
C22C1019	PRMMSG and BUF/ANSLIST not allowed
C22C101A	Buffer list not double-word aligned
C22C101B	Answer list not double-word aligned
C22C101C	No control buffer exists
C22C2030	Function not supported for CSS
C22C20nn	IUCV reply error base code
C22C2001	Invalid path ID
C22C2002	Path quiesced
C22C2003	Message limit exceeded
C22C2004	Priority message not allowed on this path
C22C2005	Buffer too short for message
C22C2006	Fetch protection exception
C22C2007	Addressing exception
C22C2008	Class or path invalid
C22C2009	Message has been purged

<b>Four-Byte Code</b>	<b>Description</b>
C22C200A	Message length negative
C22C200B	Target is not in system
C22C200C	Target is not prepared for IUCV
C22C200D	Invoker max connect count exceeded
C22C200E	Target max connect count exceeded
C22C200F	Not authorized to communicate with target
C22C2010	Invalid system service name
C22C2011	Invalid function code
C22C2012	Invalid message limit
C22C2013	Duplicate buffer declaration
C22C2014	Path has been severed
C22C2015	Parm list message not allowed
C22C2016	Send list invalid
C22C2017	Negative length in list
C22C2018	Invalid total list length
C22C2019	PRMMSG and BUF/ ANSLIST not allowed
C22C201A	Buffer list not double-word aligned
C22C201B	Answer list not double-word aligned
C22C201C	No control buffer exists
C22C3030	Function not supported for CSS
C22C30nn	IUCV sever error base code
C22C3001	Invalid path ID
C22C3002	Path quiesced
C22C3003	Message limit exceeded
C22C3004	Priority message not allowed on this path
C22C3005	Buffer too short for message
C22C3006	Fetch protection exception
C22C3007	Addressing exception
C22C3008	Class or path invalid

Four-Byte Code	Description
C22C3009	Message has been purged
C22C300A	Message length negative
C22C300B	Target is not in system
C22C300C	Target is not prepared for IUCV
C22C300D	Invoker max connect count exceeded
C22C300E	Target max connect count exceeded
C22C300F	Not authorized to communicate with target
C22C3010	Invalid system service name
C22C3011	Invalid function code
C22C3012	Invalid message limit
C22C3013	Duplicate buffer declaration
C22C3014	Path has been severed
C22C3015	Parm list message not allowed
C22C3016	Send list invalid
C22C3017	Negative length in list
C22C3018	Invalid total list length
C22C3019	PRMMSG and BUF/ANSLIST not allowed
C22C301A	Buffer list not double-word aligned
C22C301B	Answer list not double-word aligned
C22C301C	No control buffer exists
C22C3030	Function not supported for CSS

## C22Dxxxx T01 ISIO

The following table lists X'C22D' instance codes.

Four-Byte Code	Description
C22D0001	IUCV flags invalid
C22D0002	Answer area too small
C22D0003	Option data too long
C22D0011	IUCV flags invalid
C22D0012	Answer area too small (file control)
C22D0013	Invalid command value
C22D0014	Invalid argument value
C22D0080	Authorization violation
C22D00E1	Unable to obtain SAW
C22D00E2	Unable to obtain MBUF
C22D00FF	IUCV not available
C22D01nn	IUCV receive error base code
C22D0101	Invalid path ID
C22D0102	Path quiesced
C22D0103	Message limit exceeded
C22D0104	Priority message not allowed on this path
C22D0105	Buffer too short for message
C22D0106	Fetch protection exception
C22D0107	Addressing exception
C22D0108	Class or path invalid
C22D0109	Message has been purged
C22D010A	Message length negative
C22D010B	Target is not in system
C22D010C	Target is not prepared for IUCV
C22D010D	Invoker max connect count exceeded
C22D010E	Target max connect count exceeded

Four-Byte Code	Description
C22D010F	Not authorized to communicate with target
C22D0110	Invalid system service name
C22D0111	Invalid function code
C22D0112	Invalid message limit
C22D0113	Duplicate buffer declaration
C22D0114	Path has been severed
C22D0115	Parm list message not allowed
C22D0116	Send list invalid
C22D0117	Negative length in list
C22D0118	Invalid total list length
C22D0119	PRMMSG and BUF/ ANSLIST not allowed
C22D011A	Buffer list not double-word aligned
C22D011B	Answer list not double-word aligned
C22D011C	No control buffer exists
C22D0130	Function not supported for CSS
C22D02nn	IUCV reply error base code
C22D0201	Invalid path ID
C22D0202	Path quiesced
C22D0203	Message limit exceeded
C22D0204	Priority message not allowed on this path
C22D0205	Buffer too short for message
C22D0206	Fetch protection exception
C22D0207	Addressing exception
C22D0208	Class or path invalid
C22D0209	Message has been purged
C22D020A	Message length negative
C22D020B	Target is not in system
C22D020C	Target is not prepared for IUCV
C22D020D	Invoker max connect count exceeded

Four-Byte Code	Description
C22D020E	Target max connect count exceeded
C22D020F	Not authorized to communicate with target
C22D0210	Invalid system service name
C22D0211	Invalid function code
C22D0212	Invalid message limit
C22D0213	Duplicate buffer declaration
C22D0214	Path has been severed
C22D0215	Parm list message not allowed
C22D0216	Send list invalid
C22D0217	Negative length in list
C22D0218	Invalid total list length
C22D0219	PRMMSG and BUF/ANSLIST not allowed
C22D021A	Buffer list not double-word aligned
C22D021B	Answer list not double-word aligned
C22D021C	No control buffer exists
C22D0230	Function not supported for CSS
C22D03nn	Get Latch error base code
C22D0304	Latch held exclusive, request was exclusive
C22D0308	Logic error
C22D030C	Bad parameter
C22D0310	User held the CML
C22D0314	Suspend failed
C22D0318	Storage obtain failed
C22D031C	SFRE entry in use
C22D0318	Storage obtain failed
C22D0320	Latch held shared, request was exclusive
C22D0324	Latch held exclusive, request was shared
C22D04nn	Free Latch error base code

Four-Byte Code	Description
C22D0404	Latch not held
C22D0408	Logic error
C22D040C	Bad parameter

## C22Exxxx T011SLI

The following table lists X'C22E' instance codes.

Four-Byte Code	Description
C22E0001	IUCV not available
C22E0002	Unable to obtain SAW
C22E0003	IUCV header error
C22E0004	Invalid socket type
C22E01nn	IUCV error base code if IPRCODE was saved
C22E0101	Invalid path ID
C22E0102	Path quiesced
C22E0103	Message limit exceeded
C22E0104	Priority message not allowed on this path
C22E0105	Buffer too short for message
C22E0106	Fetch protection exception
C22E0107	Addressing exception
C22E0108	Class or path invalid
C22E0109	Message has been purged
C22E010A	Message length negative
C22E010B	Target is not in system
C22E010C	Target is not prepared for IUCV
C22E010D	Invoker max connect count exceeded
C22E010E	Target max connect count exceeded



Four-Byte Code	Description
C22E010F	Not authorized to communicate with target
C22E0110	Invalid system service name
C22E0111	Invalid function code
C22E0112	Invalid message limit
C22E0113	Duplicate buffer declaration
C22E0114	Path has been severed
C22E0115	Parm list message not allowed
C22E0116	Send list invalid
C22E0117	Negative length in list
C22E0118	Invalid total list length
C22E0119	PRMMMSG and BUF/ANSLIST not allowed
C22E011A	Buffer list not double-word aligned
C22E011B	Answer list not double-word aligned
C22E011C	No control buffer exists
C22E0130	Function not supported for CSS
C22E02nn	IUCV error base code if IPRCODE was not saved
C22E0201	Invalid path ID
C22E0202	Path quiesced
C22E0203	Message limit exceeded
C22E0204	Priority message not allowed on this path
C22E0205	Buffer too short for message
C22E0206	Fetch protection exception
C22E0207	Addressing exception
C22E0208	Class or path invalid
C22E0209	Message has been purged
C22E020A	Message length negative
C22E020B	Target is not in system
C22E020C	Target is not prepared for IUCV
C22E020D	Invoker max connect count exceeded

Four-Byte Code	Description
C22E020E	Target max connect count exceeded
C22E020F	Not authorized to communicate with target
C22E0210	Invalid system service name
C22E0211	Invalid function code
C22E0212	Invalid message limit
C22E0213	Duplicate buffer declaration
C22E0214	Path has been severed
C22E0215	Parm list message not allowed
C22E0216	Send list invalid
C22E0217	Negative length in list
C22E0218	Invalid total list length
C22E0219	PRMMSG and BUF/ANSLIST not allowed
C22E021A	Buffer list not double-word aligned
C22E021B	Answer list not double-word aligned
C22E021C	No control buffer exists
C22E0230	Function not supported for CSS
C22E03nn	Get latch failure base code
C22E0304	Latch held exclusive, request was exclusive
C22E0308	Logic error
C22E030C	Bad parameter
C22E0310	User held the CML
C22E0314	Suspend failed
C22E0318	Storage obtain failed
C22E031C	SFRE entry in use
C22E0318	Storage obtain failed
C22E0320	Latch held shared, request was exclusive
C22E0324	Latch held exclusive, request was shared
C22E04nn	Free latch failure base code
C22E0404	Latch not held

Four-Byte Code	Description
C22E0408	Logic error
C22E040C	Bad parameter
C22E0999	IUCV purged message

## C22Fxxxx T011SRV

The following table lists X'C22F' instance codes.

Four-Byte Code	Description
C22F0000	Return to caller pending callback
C22F0010	TRGCLS is invalid
C22F0020	IUCV flags invalid - READ
C22F0023	IUCV answer length is negative - READ
C22F0024	Socket flags set for READ
C22F0030	IUCV flags invalid - RECVFROM
C22F0033	IUCV answer length is negative - RECVFROM
C22F0034	Socket flags set for RECVFROM
C22F0050	Attempting recovery after IUCV error
C22F0060	Receive data was queued
C22F00E1	Unable to obtain primary SAW
C22F00E4	Unable to obtain MBUF
C22F00FF	IUCV not available
C22F01nn	IUCV error base code
C22F0101	Invalid path ID
C22F0102	Path quiesced
C22F0103	Message limit exceeded
C22F0104	Priority message not allowed on this path
C22F0105	Buffer too short for message

Four-Byte Code	Description
C22F0106	Fetch protection exception
C22F0107	Addressing exception
C22F0108	Class or path invalid
C22F0109	Message has been purged
C22F010A	Message length negative
C22F010B	Target is not in system
C22F010C	Target is not prepared for IUCV
C22F010D	Invoker max connect count exceeded
C22F010E	Target max connect count exceeded
C22F010F	Not authorized to communicate with target
C22F0110	Invalid system service name
C22F0111	Invalid function code
C22F0112	Invalid message limit
C22F0113	Duplicate buffer declaration
C22F0114	Path has been severed
C22F0115	Parm list message not allowed
C22F0116	Send list invalid
C22F0117	Negative length in list
C22F0118	Invalid total list length
C22F0119	PRMMSG and BUF/ANSLIST not allowed
C22F011A	Buffer list not double-word aligned
C22F011B	Answer list not double-word aligned
C22F011C	No control buffer exists
C22F0130	Function not supported for CSS
C22F02nn	Get Latch error base code - normal processing
C22F0204	Latch held exclusive, request was exclusive
C22F0208	Logic error
C22F020C	Bad parameter
C22F0210	User held the CML

Four-Byte Code	Description
C22F0214	Suspend failed
C22F0218	Storage obtain failed
C22F021C	SFRE entry in use
C22F0218	Storage obtain failed
C22F0220	Latch held shared, request was exclusive
C22F0224	Latch held exclusive, request was shared
C22F03nn	Get Latch error base code - resumed processing
C22F0304	Latch held exclusive, request was exclusive
C22F0308	Logic error
C22F030C	Bad parameter
C22F0310	User held the CML
C22F0314	Suspend failed
C22F0318	Storage obtain failed
C22F031C	SFRE entry in use
C22F0318	Storage obtain failed
C22F0320	Latch held shared, request was exclusive
C22F0324	Latch held exclusive, request was shared
C22F05nn	Get Latch error base code - error recovery
C22F0504	Latch held exclusive, request was exclusive
C22F0508	Logic error
C22F050C	Bad parameter
C22F0510	User held the CML
C22F0514	Suspend failed
C22F0518	Storage obtain failed
C22F051C	SFRE entry in use
C22F0518	Storage obtain failed
C22F0520	Latch held shared, request was exclusive
C22F0524	Latch held exclusive, request was shared
C22F06nn	Free Latch error base code - normal processing

Four-Byte Code	Description
C22F0604	Latch not held
C22F0608	Logic error
C22F060C	Bad parameter
C22F07nn	Free Latch error base code - before suspend
C22F0704	Latch not held
C22F0708	Logic error
C22F070C	Bad parameter
C22F08nn	Free Latch error base code - error recovery
C22F0804	Latch not held
C22F0808	Logic error
C22F080C	Bad parameter
C22F09nn	Free Latch error base code - receive data queued
C22F0904	Latch not held
C22F0908	Logic error
C22F090C	Bad parameter
C22F1FFF	Bulk mode - not supported yet

## C230xxxxT011SSD

The following table listsX'C230' instance codes.

Four-Byte Code	Description
C2300001	Invalid IUCV flags
C2300002	ANSWER area too small
C2300003	REQUEST area too small
C2300020	Invalid IUCV Target Class
C2300023	Invalid flags
C23000E1	Unable to obtain a SAW
C23000FF	PC number is zero
C2300100	Error on IUCV RECEIVE
C2300500	Error on IUCV REPLY
C2300700	Unable to obtain SEPM LATCH
C2300900	Unable to obtain SEPM LATCH
C2300B00	Error freeing SEPM LATCH
C2300C00	Error freeing SEPM LATCH
C2300D00	Error freeing SEPM LATCH
C2300E00	Error freeing SEPM LATCH
C2301FFF	Bulk Mode not supported

## C231xxxxT011SSH

The following table lists X'C231' instance codes.

Four-Byte Code	Description
C2310001	IUCV not active
C2310002	Error obtaining SAW
C2310003	Invalid IUCV flags
C2310100	Return Code 1 from IUCV function
C2310200	Return Code other than 1 from IUCV function
C2310300	Unable to obtain SEPM LATCH
C2310400	Error freeing SEPM LATCH
C2310999	Message Purged

## C232xxxxT011SSK

The following table lists X'C232' instance codes.

Four-Byte Code	Description
C2320001	IUCV not active
C2320002	Error obtaining SAW
C2320003	Invalid IUCV flags
C2320004	IUCV header error
C2320005	Socket number out of range
C2320006	IUCV Socket Array Pointer error (INDEX4)
C2320007	Socket number already in use
C2320008	Unable to obtain an SEPM block
C2320009	Domain not supported (must be AF_INET)
C232000A	Invalid socket type
C232000B	Invalid socket protocol



Four-Byte Code	Description
C232000C	Invalid socket type
C232000D	Socket type RAW and protocol TCP
C232000E	Socket type STREAM and protocol not TCP
C232000F	Socket type DATAGRAM and protocol not UDP
C2320010	Socket request exceeds maximum sockets allowed.
C2320011	Socket refused, Unicenter TCPAccess is shutting down.
C2320012	TCP/IP stack Short-On-Storage (SOS)
C2320100	Return Code 1 from IUCV function
C2320200	Return Code other than 1 from IUCV function
C2320999	Message Purged

## C233xxxxT011SSL

The following table lists X'C233' instance codes.

Four-Byte Code	Description
C2330001	IUCV not active
C2330002	Error obtaining SAW
C2330003	Invalid IUCV flags
C2330004	IUCV header error
C2330005	Error obtaining MBUF
C2330006	No sockets selected & no timeout value
C2330007	Error obtaining secondary SAWs
C2330008	Error obtaining secondary SAWs (fewer returned)
C2330100	Return Code 1 from IUCV function
C2330200	Return Code other than 1 from IUCV function
C2330300	Unable to obtain SEPM LATCH
C2330400	Unable to obtain SEPM LATCH

Four-Byte Code	Description
C2330500	Unable to obtain IUCV Control Latch
C2330600	Unable to obtain IUCV Control Latch
C2330700	Unable to obtain IUCV Control Latch
C2330800	Error freeing SEPM LATCH
C2330900	Error freeing IUCV Control Latch
C2330A00	Error freeing IUCV Control Latch
C2330B00	Error freeing IUCV Control Latch
C2330C00	Error freeing IUCV Control Latch
C2330D00	Error freeing SEPM LATCH
C2330EEE	Message Purged

## C234xxxxT011SSO

The following table lists X'C234' instance codes.

Four-Byte Code	Description
C2340001	Invalid IUCV flags
C2340002	ANSWER area too small
C2340003	Option data too long
C2340010	Not SOL_SOCKET for IBM options
C23400E1	Error obtaining SAW
C23400FF	PC number is zero
C2340100	Error on IUCV RECEIVE
C2340200	Error on IUCV REPLY
C2340300	Unable to obtain SEPM LATCH
C2340400	Error freeing SEPM LATCH

## C235xxxxT011SSP

The following table lists X'C235' instance codes.

Four-Byte Code	Description
C235000A	Error obtaining primary SAW
C235000B	Invalid PATHID
C2350014	Error obtaining working SAW
C2350100	IUCV SEVER failed
C2350200	Unable to obtain IUCV Control Latch
C2350300	Unable to obtain SEPM LATCH
C2350600	Error freeing SEPM LATCH
C2350700	Error freeing IUCV Control Latch
C2350E00	IUCV not active
C2350F00	PC number is zero

## C236xxxxT011STS

The following table lists X'C236' instance codes.

Four-Byte Code	Description
C2360001	IUCV not active
C2360002	Error obtaining SAW
C2360003	Invalid IUCV flags
C2360004	IUCV header error
C2360005	Socket number out of range
C2360006	IUCV Socket Array Pointer error (INDEX4)
C2360007	Socket number already in use
C2360008	No existing PATH Table
C2360009	Invalid Socket descriptor

Four-Byte Code	Description
C236000A	Socket number already in use
C236000B	Socket not being passed
C236000C	Socket not given to taker (JOBNAME)
C236000D	Socket not given to taker (AUTH)
C236000E	Error in slot in Socket Array
C2360100	Return Code 1 from IUCV function
C2360200	Return Code other than 1 from IUCV function
C2360300	Unable to obtain SEPM LATCH
C2360400	Unable to obtain IUCV Control Latch
C2360500	Error freeing SEPM LATCH
C2360600	Error freeing SEPM LATCH
C2360700	Error freeing IUCV Control Latch
C2360800	Error freeing IUCV Control Latch
C2360999	Message Purged
C2360A00	Error freeing IUCV Control Latch

## C240xxxx (01xx) T012SPC

The following table lists X'C240' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2400001	0101	XWA not found
C2400002	0102	SAVT not found
C2400003	0103	SAVX not found
C2400004	0104	SPCB not found
C2400005	0105	TSVT not found
C2400006	0106	Caller has an active FRR
C2400007	0107	ASID out of range
C2400008	0108	Unable to obtain CSQB
C2400009	0109	Socket function out of range
C240000A	010A	Unsupported socket function
C240000B	010B	System is terminating
C24000FF	01FF	Recovery SEPM not found

## C241xxxx (03xx) T012TACC

The following table lists X'C241' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2410001	0301	Unexpected abend
C2410002	0302	Recursive error
C2410003	0303	Error copying TPL to secondary
C2410004	0304	Error copying exit codes to TPL
C2410005	0305	VTPL abend
C2410020	0320	SAW not obtained
C2410021	0321	SEPM state invalid
C2410022	0322	TPLSEQNO zero
C2410023	0323	TPLOPLEN not zero
C2410024	0324	New endpoint TCEP storage invalid
C2410025	0325	New endpoint SEPM not disabled
C2410026	0326	SEPM not on listen queue

## C242xxxx (04xx) T012TADR

The following table lists X'C242' instance codes.

Two-Byte Code	Two-Byte Code	Description
C2420001	0401	Unexpected abend
C2420002	0402	Recursive error
C2420003	0403	Error copying TPL to secondary
C2420004	0404	Error copying exit codes to TPL
C2420005	0405	VTPL abend
C2420006	0406	Abend copying TPA to secondary
C2420020	0420	SAW not obtained
C2420021	0421	Data area length zero
C2420022	0422	Data area too small and NOTRUNC specified
C2420023	0423	Remote endpoint state error
C2420024	0424	Local endpoint state error

## C243xxxx (05xx) T012TBIN

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2430001	0501	Unexpected abend
C2430002	0502	Recursive error
C2430003	0503	Error copying TPL to secondary
C2430004	0504	Error copying exit codes to TPL
C2430005	0505	VTPL abend
C2430020	0520	SAW not obtained
C2430021	0521	TLI state invalid
C2430022	0522	TCP QLISTEN value not negotiable
C2430023	0523	RAW socket has non-zero QLISTEN value
C2430024	0524	UDP QLISTEN value not negotiable
C2430025	0525	TPA not big enough and OPTCD=ASSIG specified
C2430026	0526	TPA not big enough and OPTCD=NOTRUNC specified
C2430027	0527	TPA domain invalid
C2430028	0528	RAW socket missing host IP address
C2430029	0529	TCP/UDP socket missing port number
C243002A	052A	SEPM state invalid
C243002B	052B	SEPM not obtained
C243002C	052C	Port assignment failed
C243002D	052D	Port in use and REUSEADDR not set



## C244xxxx (06xx) T012TCLR

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2440001	0601	Unexpected abend
C2440002	0602	Recursive error
C2440003	0603	Error copying TPL to secondary
C2440004	0604	Error copying exit codes to TPL
C2440005	0605	VTPL abend
C2440020	0620	SAW not obtained
C2440021	0621	SEPM state invalid

## C245xxxx (07xx) T012TCLS

The following table lists X'C245' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2450001	0701	Unexpected ABEND
C2450002	0702	Recursive error
C2450003	0703	Error copying TPL to secondary
C2450004	0704	Error copying exit codes to TPL
C2450005	0705	VTPL abend
C2450020	0720	SAW not obtained
C2450021	0721	ASCB validation failed
C2450022	0722	TCB address is negative
C2450023	0723	31-bit TCB address passed
C2450024	0724	Close-pass: Close-pass pending
C2450025	0725	Close-pass: Accept pending
C2450026	0726	Close-pass: Receive pending
C2450027	0727	Close-pass: Send pending
C2450028	0728	Close-pass: Select pending

Four-Byte Code	Two-Byte Code	Description
C2450029	0729	Close-pass: MBUFs awaiting send
C245002A	072A	Close-pass: UDP/RAW send pending
C245002B	072B	Close-pass: Shutdown pending
C245002C	072C	Close-pass: Open-old retracted
C245002D	072D	Close-pass: Close Pending
C245002E	072E	Open-old: Authorization failed
C245002F	072F	SEPM state NULL after suspend/callback
C2450030	0730	SEPM state DEAD after suspend/callback
C2450031	0731	Open-old: Failed by close-delete
C2450032	0732	TCP is terminating: SSIT is unavailable
C2450033	0733	TCP is terminating: SAVT is unavailable
C2450034	0734	TCP is terminating: SAVX is unavailable
C2450035	0735	TCP is terminating: SAVXLTCH is unavailable
C2450036	0736	Close-delete already in progress

## C246xxxx (08xx) T012TCNF

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2460001	0801	Unexpected abend
C2460002	0802	Recursive error
C2460003	0803	Error copying TPL to secondary
C2460004	0804	Error copying exit codes to TPL
C2460005	0805	VTPL abend
C2460006	0806	Abend copying TPA to secondary
C2460020	0820	SAW not obtained
C2460021	0821	SEPM state invalid
C2460022	0822	Connection not done or in flight
C2460023	0823	SEPM state NULL after suspend/callback
C2460024	0824	SEPM state DEAD after suspend/callback
C2460025	0825	SEPM not connected
C2460026	0826	User TPA area is too small and OPTCD=NOTRUNC specified

## C247xxxx (09xx)T012TCON

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2470001	0901	Unexpected abend
C2470002	0902	Recursive error
C2470003	0903	Error copying TPL to secondary
C2470004	0904	Error copying exit codes to TPL
C2470005	0905	VTPL abend
C2470020	0920	Not enough memory to perform function
C2470021	0921	No SAWs obtained
C2470022	0922	Only one SAW obtained
C2470023	0923	SEPM state invalid
C2470024	0924	Protocol address area is too small
C2470025	0925	Invalid internet domain
C2470026	0926	Port number is zero
C2470027	0927	Attempting to replace non-dead SEPM
C2470028	0928	Unable to obtain SEPM

## C248xxxx (0Axx) T012TDIS

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2480001	0A01	Unexpected abend
C2480002	0A02	Recursive error
C2480003	0A03	Error copying TPL to secondary
C2480004	0A04	Error copying exit codes to TPL
C2480005	0A05	VTPL abend
C2480020	0A20	SAW not obtained
C2480021	0A21	SEPM state invalid
C2480022	0A22	Socket state dead/ failed and TPL OPTCD=ABORT specified

## C249xxxx (0Bxx) T012TINF

The following table lists X'C249' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2490001	0B01	Unexpected abend
C2490002	0B02	Recursive error
C2490003	0B03	Error copying TPL to secondary
C2490004	0B04	Error copying exit codes to TPL
C2490005	0B05	VTPL abend
C2490020	0B20	SAW not obtained
C2490021	0B21	Data area too small and OPTCD=NOTRUNC specified

## C24Axxxx (0Cxx) T012TLIS

The following table lists X'C24A' instance codes.

Four-Byte Code	Two-Byte Code	Description
C24A0001	0C01	Unexpected abend
C24A0002	0C02	Recursive error
C24A0003	0C03	Error copying TPL to secondary
C24A0004	0C04	Error copying exit codes to TPL
C24A0005	0C05	VTPL abend
C24A0006	0C06	Abend copying TPA to secondary
C24A0020	0C20	SAW not obtained
C24A0021	0C21	SEPM state invalid
C24A0022	0C22	SEPM state NULL after suspend/callback
C24A0023	0C23	SEPM state DEAD after suspend/callback
C24A0024	0C24	Transaction would block and OPTCD=NOBLOCK specified
C24A0025	0C25	Child SEPM is unavailable
C24A0026	0C26	User TPA area is too small and OPTCD=NOTRUNC specified
C24A0027	0C27	User attempted second TLISTEN with same TPL

## C24Bxxxx (0Dxx) T012TOPN

The following table lists X'C24B' instance codes.

Four-Byte Code	Two-Byte Code	Description
C24B0000	0D00	Use SAW exit codes
C24B0001	0D01	Unexpected abend
C24B0002	0D02	Recursive error
C24B0003	0D03	Error copying TPL to secondary
C24B0004	0D04	Error copying exit codes to TPL
C24B0005	0D05	VTPL abend
C24B0006	0D06	Abend reading TUB storage
C24B0007	0D07	Abend reading ACEE storage
C24B0020	0D20	No memory (IMEMCHK)
C24B0021	0D21	No SAW (IGPOL)
C24B0022	0D22	Invalid End Point Id (Optcd=New)
C24B0023	0D23	ASCB validation failed
C24B0024	0D24	Invalid End Point Id (OPTCD=OLD) (No ATCB)
C24B0025	0D25	Invalid End Point Id (OPTCD=OLD) (No SEPM)
C24B0026	0D26	Close Pass SEPM in invalid state
C24B0027	0D27	Open Old already occurred on target endpoint
C24B0028	0D28	Close Pass ASCB Auth Check failed
C24B0029	0D29	Close Pass TCB Auth Check failed
C24B002A	0D2A	TUB above the line, APCB opened in 24-bit mode
C24B002B	0D2B	TUB user ID is too long
C24B002C	0D2C	TUB group name is too long
C24B002D	0D2D	TUB password is too long
C24B002E	0D2E	OPTCD=ACEE, but ACEE pointer is zero
C24B002F	0D2F	ACEE above the line, APCB opened in 24-

Four-Byte Code	Two-Byte Code	Description
		bit mode
C24B0030	0D30	ACEE failed validation, control block ID
C24B0031	0D31	ACEE failed validation, zero length
C24B0032	0D32	Invalid TPL Domain
C24B0033	0D33	Invalid TPL Type
C24B0034	0D34	Unable to locate Aopen ATCB
C24B0035	0D35	Unable to allocate SEPM
C24B0036	0D36	Unable to allocate TTCB
C24B0037	0D37	Unable to allocate IRB / IQE
C24B0038	0D38	Unable to allocate Savearea
C24B0039	0D39	Unable to allocate TCEP
C24B003A	0D3A	Invalid TXL length
C24B003B	0D3B	T01XCREA call gave return code four
C24B003C	0D3C	T01XCREA call gave return code eight
C24B003D	0D3D	SEPM state NULL after suspend/callback
C24B003E	0D3E	SEPM state DEAD after suspend/callback
C24B003F	0D3F	Close Pass SEPM freed before callback/resume
C24B0040	0D40	AOPEN in 24-bit mode; exit/ECB in 31- bit mode
C24B0041	0D41	ECB/exit not on halfword boundary
C24B0042	0D42	ECB not on fullword boundary



## C24Cxxxx (0Exx) T012TOPT

The following table lists X'C24C' instance codes.

Four-Byte Code	Two-Byte Code	Description
C24C0001	0E01	Unexpected abend
C24C0002	0E02	Recursive error
C24C0003	0E03	Error copying TPL to secondary
C24C0004	0E04	Error copying exit codes to TPL
C24C0005	0E05	VTPL abend
C24C0006	0E06	Error during rollback
C24C0020	0E20	SAW not obtained
C24C0021	0E21	TLI state invalid
C24C0022	0E22	No option buffer provided
C24C0023	0E23	Option buffer length not positive
C24C0024	0E24	Option length not positive
C24C0025	0E25	OPTCD=API and option number negative
C24C0026	0E26	OPTCD=API and option number out of range
C24C0027	0E27	OPTCD=TP and option number not positive
C24C0028	0E28	Option number out of range
C24C0029	0E29	Option not supported
C24C002A	0E2A	Option length greater than maximum allowed
C24C002B	0E2B	OPTCD=VERIFY not supported (IOCTL)
C24C002C	0E2C	OPTCD=DEFAULT not supported (IOCTL)
C24C002D	0E2D	OPTCD=DECLARE not supported
C24C002E	0E2E	OPTCD=DECLARE user length too large
C24C002F	0E2F	T01XOPT1 ended in error
C24C0030	0E30	OPTCD=VERIFY not supported
C24C0031	0E31	OPTCD=VERIFY user length too large

Four-Byte Code	Two-Byte Code	Description
C24C0032	0E32	Unknown exception from T01XOPT1
C24C0033	0E33	OPTCD=DEFAULT not supported
C24C0034	0E34	OPTCD=DEFAULT user length too large
C24C0035	0E35	OPTCD=DECLARE not supported (IOCTL)
C24C0036	0E36	OPTCD=DECLARE user length too large (IOCTL)
C24C0037	0E37	IOCTL OPTCD=DECLARE and not HASN=SASN or sup
C24C0038	0E38	SET INTERFACE METRIC not in IFS address space
C24C0039	0E39	Option length greater than maximum allowed
C24C003A	0E3A	OPTCD=QUERY not supported
C24C003B	0E3B	OPTCD=QUERY not supported (IOCTL)
C24C003C	0E3C	OPTCD=QUERY user length too large (IOCTL)
C24C003D	0E3D	Unable to get MBUF

## C24Dxxxx (0Fxx) T012TRCV

The following table lists X'C24D' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2400001	0F01	Unexpected abend
C2400002	0F02	Recursive error
C2400003	0F03	Error copying TPL to secondary
C2400004	0F04	Error copying exit codes to TPL
C2400005	0F05	VTPL abend
C2400006	0F06	Abend verifying vectors
C2400007	0F07	Abend in T01XREAD
C2400008	0F08	Abend in T012SUSP
C2400009	0F09	Abend in T01SMOVE
C240000A	0F0A	Abend adjusting vector lengths
C2400020	0F20	SAW not obtained
C2400021	0F21	SEPM state invalid
C2400022	0F22	No protocol options buffer provided
C2400023	0F23	Protocol options buffer length not eight
C2400024	0F24	Timeout option length not eight
C2400025	0F25	Timeout value too high
C2400026	0F26	OPTCD=INDIR vector length not a multiple of eight
C2400027	0F27	TRECV queue count exceeded
C2400028	0F28	Function would block and OPTCD=NOBLOCK specified
C2400029	0F29	Unable to set SEPM timer
C240002A	0F2A	No SAWs queued after suspend/callback
C240002B	0F2B	SEPM state NULL after suspend/callback
C240002C	0F2C	SEPM state DEAD after suspend/callback
C240002D	0F2D	Unable to set SEPM timer
C240002E	0F2E	Data requested, but none received

## C24Exxxx (10xx) T012TRER

The following table lists X'C24E' instance codes.

Four-Byte Code	Two-Byte Code	Description
C24E0001	1001	Unexpected abend
C24E0002	1002	Recursive error
C24E0003	1003	Error copying TPL to secondary
C24E0004	1004	Error copying exit codes to TPL
C24E0005	1005	VTPL abend
C24E0006	1006	Abend copying protocol address to TPA
C24E0020	1020	SAW not obtained
C24E0021	1021	Socket type not connectionless
C24E0022	1022	Remote endpoint state error
C24E0023	1023	Local endpoint state error
C24E0024	1024	Address area too small and OPTCD=NOTRUNC specified
C24E0025	1025	Child SEPM is unavailable
C24E0026	1026	User TPA area is too small and OPTCD=NOTRUNC specified

## C24Fxxxx (11xx) T012TRFR

The following table lists X'C24F' instance codes.

Four-Byte Code	Two-Byte Code	Description
C24F0001	1101	Unexpected abend
C24F0002	1102	Recursive error
C24F0003	1103	Error copying TPL to secondary
C24F0004	1104	Error copying exit codes to TPL
C24F0005	1105	VTPL abend
C24F0006	1106	Abend verifying vectors
C24F0007	1107	Abend in T01XREAD
C24F0008	1108	Abend in T012SUSP
C24F0009	1109	Abend in T01SMOVE
C24F000A	110A	Abend copying TPA
C24F000B	110B	Abend adjusting vector lengths
C24F0020	1120	SAW not obtained
C24F0021	1121	OPTCD=INDIR vector length not a multiple of eight
C24F0022	1122	SAW on send queue has shutdown function
C24F0023	1123	SEPM state invalid
C24F0024	1124	TSEND queue count exceeded
C24F0025	1125	Function would block and mode is TLI
C24F0026	1126	Function would block and OPTCD=NOBLOCK specified
C24F0027	1127	No SAWs queued after suspend/callback
C24F0028	1128	SEPM state NULL after suspend/callback
C24F0029	1129	SEPM state DEAD after suspend/callback
C24F002A	112A	MBUFs not obtained (T01SMBUF return code four)
C24F002B	112B	MBUF byte count not positive (T01SMBUF return code eight)

Four-Byte Code	Two-Byte Code	Description
C24F002C	112C	Secondary SAW not obtained
C24F002D	112D	MBUFs not obtained (T01SMBUF return code four)
C24F002E	112E	MBUF byte count not positive (T01SMBUF return code eight)

## C250xxxx (12xx) T012TRJT

The following table lists X'C250' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2500001	1201	Unexpected abend
C2500002	1202	Recursive error
C2500003	1203	Error copying TPL to secondary
C2500004	1204	Error copying exit codes to TPL
C2500005	1205	VTPL abend
C2500020	1220	SAW not obtained
C2500021	1221	SEPM state error
C2500022	1222	TPL sequence number zero
C2500023	1223	SEPM not on TLISTEN queue or queue blocked
C2500024	1224	SEPM not connected and TPL specified ABORT

## C251xxxx (13xx) T012TRLK

The following table lists X'C251' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2510001	1301	Unexpected abend
C2510002	1302	Recursive error
C2510003	1303	Error copying TPL to secondary
C2510004	1304	Error copying exit codes to TPL
C2510005	1305	VTPL abend
C2510020	1320	SAW not obtained
C2510021	1321	SEPM state invalid
C2510022	1322	Release in progress and reset received
C2510023	1323	Receive queue blocked
C2510024	1324	SEPM state NULL after suspend/callback
C2510025	1325	SEPM state DEAD after suspend/callback
C2510026	1326	SEPM state invalid after T01XSHT1

## C252xxxx (14xx) T012TRLS

The following table lists X'C252' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2520001	1401	Unexpected abend
C2520002	1402	Recursive error
C2520003	1403	Error copying TPL to secondary
C2520004	1404	Error copying exit codes to TPL
C2520005	1405	VTPL abend
C2520006	1406	Abend copying TPA to secondary
C2520020	1420	SAW not obtained
C2520021	1421	SEPM state invalid
C2520022	1422	SEPM state NULL after suspend/callback
C2520023	1423	SEPM state DEAD after suspend/callback
C2520024	1424	SEPM state invalid after T01XSHT1

## C253xxxx (15xx) T012TRCT

The following table lists X'C253' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2530001	1501	Unexpected abend
C2530002	1502	Recursive error
C2530003	1503	Error copying TPL to secondary
C2530004	1504	Error copying exit codes to TPL
C2530005	1505	VTPL abend
C2530020	1520	SAW not obtained
C2530021	1521	SEPM state error
C2530022	1522	SEPM accept queue empty/blocked



## C254xxxx (16xx) T012TSND

The following table lists X'C254' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2540001	1601	Unexpected abend
C2540002	1602	Recursive error
C2540003	1603	Error copying TPL to secondary
C2540004	1604	Error copying exit codes to TPL
C2540005	1605	VTPL abend
C2540006	1606	Abend verifying indirect data buffers
C2540007	1607	Abend in T01XSEND
C2540008	1608	Abend in T012SUSP
C2540009	1609	Abend getting MBUFs
C254000A	160A	Abend getting secondary SAWs
C254000B	160B	Abend getting MBUFs
C254000C	160C	Abend in T01SMOVE
C254000D	160D	Abend in T01XSND2
C254000E	160E	Abend updating tplCOUNT
C2540020	1620	SAW not obtained
C2540021	1621	OPTCD=INDIR vector length not a multiple of eight
C2540022	1622	SAW on send queue has shutdown function
C2540023	1623	SEPM state invalid
C2540024	1624	TSEND queue count exceeded
C2540025	1625	Function would block and mode is TLI
C2540026	1626	Function would block and OPTCD=NOBLOCK specified
C2540027	1627	Table entry available
C2540028	1628	SEPM state NULL after suspend/callback
C2540029	1629	SEPM state DEAD after suspend/callback
C254002A	162A	MBUFs not obtained (T01SMBUF return

Four-Byte Code	Two-Byte Code	Description
		code four)
C254002B	162B	MBUF byte count not positive (T01SMBUF return code eight)
C254002C	162C	Secondary SAW not obtained
C254002D	162D	MBUFs not obtained (T01SMBUF return code four)
C254002E	162E	MBUF byte count not positive (T01SMBUF return code eight)

## C255xxxx (17xx) T012TSTO

The following table lists X'C255' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2550001	1701	Unexpected abend
C2550002	1702	Recursive error
C2550003	1703	Error copying TPL to secondary
C2550004	1704	Error copying exit codes to TPL
C2550005	1705	VTPL abend
C2550006	1706	Abend verifying indirect data buffers
C2550007	1707	Abend in T01XSND1
C2550008	1708	Abend in T012SUSP
C2550009	1709	Abend getting secondary SAWs
C255000A	170A	Abend getting MBUFs
C255000B	170B	Abend in T01SMOVE
C255000C	170C	Abend in T01XSND2
C255000D	170D	Abend updating tplCOUNT
C2550020	1720	SAW not obtained
C2550021	1721	SEPM state invalid
C2550022	1722	Protocol address omitted/too small
C2550023	1723	Vector length not a multiple of eight

Four-Byte Code	Two-Byte Code	Description
C2550024	1724	TSEND queue count exceeded
C2550025	1725	Function would block and OPTCD=NOBLOCK specified
C2550026	1726	Function would block and mode is TLI
C2550027	1727	Function would block and OPTCD=NOBLOCK
C2550028	1728	TSEND queue blocked
C2550029	1729	SEPM state NULL after suspend/callback
C255002A	172A	SEPM state DEAD after suspend/callback
C255002B	172B	Secondary SAW not obtained
C255002C	172C	MBUFs not obtained (T01SMBUF return code four)
C255002D	172D	MBUF byte count not positive (T01SMBUF return code eight)

## C256xxxx (18xx) T012TUNB

The following table lists X'C256' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2560001	1801	Unexpected abend
C2560002	1802	Recursive error
C2560003	1803	Error copying TPL to secondary
C2560004	1804	Error copying exit codes to TPL
C2560005	1805	VTPL abend
C2560020	1820	SAW not obtained
C2560021	1821	SEPM state invalid

## C257xxxx (19xx) T012TUSR

The following table lists X'C257' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2570001	1901	Unexpected abend
C2570002	1902	Recursive error
C2570003	1903	Error copying TPL to secondary
C2570004	1904	Error copying exit codes to TPL
C2570005	1905	VTPL abend
C2570006	1906	TUB storage fetch error
C2570007	1907	ACEE storage fetch error
C2570020	1920	SAW not obtained
C2570021	1921	User parameter missing
C2570022	1922	User parameter 31 bit, TPL 24 bit
C2570023	1923	TUB user ID length zero
C2570024	1924	TUB user ID too big
C2570025	1925	TUB group name too big
C2570026	1926	TUB password too big
C2570027	1927	ACEE block tag error
C2570028	1928	ACEE block length error
C2570029	1929	ACEE user ID length zero
C257002A	192A	ACEE user ID too big
C257002B	192B	TCEP not available
C257002C	192C	TCEP not available, and user parm is ACEE

## C258xxxx (1Axx) T012AOPN

For more information on AOPEN/ACLOSE error codes, see AOPEN and ACLOSE Error Codes in the Chapter “API Return Codes.”

The following table lists X'C258' error codes.

<b>R00</b>	<b>apcbERRC</b>	<b>apcbDGNC</b>	<b>Description</b>
0E	-	-	Not enough storage to continue
06	-	-	APCB tag check failed/storage not writable
09	09	0004	APCB is already open
0B	-	-	APCB is busy
06	06	0006	Access method is invalid
0B	0B	0007	Environment is invalid
06	06	0008	Exit list is invalid
03	03	0009	APCB initialization failed
07	07	000A	Unable to find application ID
07	07	000B	Matching ATCB has no TTCB
0E	0E	000C	Unable to obtain ATCB
0E	0E	000D	Unable to obtain TTCB
0E	0E	000E	Unable to obtain IRBW
0E	0E	000F	Unable to obtain IRB save area
0E	0E	0010	Unable to obtain APCX
14	14	0011	OPTCD=AUTHEXIT and caller in problem state
07	07	0012	OPTCD=AUTHEXIT and MVS level less than 5.2.2
02	-	-	System is terminating

## C259xxxx (1Bxx) T012ACLS

For more information on AOPEN/ ACLOSE error codes, see AOPEN and ACLOSE Error Codes in the Chapter “API Return Codes.”

The following table lists X'C259' error codes.

R00	abcbERRC	apcbDGNC	Description
06	-	-	APCB tag check failed/storage is not writable
0A	0A	0003	APCB is already closed
0B	-	-	APCB is busy
15	15	0005	TTRV tag check failed
0D	0D	0006	TCVT tag check failed
06	06	0007	ACLOSE done on different task from AOPEN
07	07	0008	SEPM close failed
07	07	0009	APCX remove failed
01	01	0009	Subsystem not configured (set by T01PSTUB)
07	07	000A	APCB reset failed
02	02	000A	Subsystem not active (set by T01PSTUB)
07	07	000B	System is terminating
07	07	000B	Failed to free SAS/C environment (set by T01PSTUB)
07	07	000C	IRB cleanup failed.

## C25Axxxx (1Cxx) T012TCHK

The following table lists X'C243' instance codes.

Four-Byte Code	Two-Byte Code	Description
C25A0001	1C01	Unexpected abend
C25A0002	1C02	Recursive error
C25A0003	1C03	Error copying TPL to secondary
C25A0004	1C04	Error copying exit codes to TPL
C25A0005	1C05	VTPL abend
C25A0020	1C20	SAW not obtained
C25A0021	1C21	TPL is already inactive
C25A0022	1C22	Function exit not completed
C25A0023	1C23	ECB post requested in SRB mode
C25A0024	1C24	SEPM could not be deleted

## C25Bxxxx (1Dxx) T012TERR

This table lists X'C25B' instance codes.

Four-Byte Code	Two-Byte Code	Description
C25B0001	1D01	Unexpected abend
C2570006	1906	TUB storage fetch error
C25B0002	1D02	Recursive error
C25B0003	1D03	Error copying TPL to secondary
C25B0004	1D04	Error copying exit codes to TPL
C25B0005	1D05	VTPL abend
C25B0006	1D06	Abend obtaining TEM storage
C25B0007	1D07	Abend building TEM
C25B0020	1D20	SAW not obtained
C25B0021	1D21	TPL is active
C25B0022	1D22	STORAGE OBTAIN for TEM failed
C25B0023	1D23	Summary format not supported

## C25Cxxxx (1Exx) T012TSTA

The following table lists X'C25C' instance codes.

Four-Byte Code	Two-Byte Code	Description
C25C0001	1E01	Unexpected abend
C25C0002	1E02	Recursive error
C25C0003	1E03	Error copying TPL to secondary
C25C0004	1E04	Error copying exit codes to TPL
C25C0005	1E05	VTPL abend
C25C0020	1E20	SAW not obtained
C25C0021	1E21	TPL is active



## C25Fxxxx (21xx) T01PSTUB

The following table lists X'C25F' instance codes.

Four-Byte Code	Two-Byte Code	Description
C25F0001	2101	Unable to obtain TSCE
C25F0002	2102	SAS/C initialization failed
C25F0003	2103	SAVT not available
C25F0004	2104	PC not available
C25F0005	2105	TCP restarted after AOPEN
C25F0006	2106	Subsystem not active
C25F0007	2107	PC not available
C25F0008	2108	<not assigned>
C25F0009	2109	Subsystem not found
C25F000A	210A	Subsystem not active
C25F000B	210B	SAS/C termination failed
C25F000C	210C	TCP restarted after AOPEN
C25F000D	210D	User active during TCP restart

## C262xxxx (6Fxx) T012TPLK

The following table lists X'C262' instance codes.

Four-Byte Code	Two-Byte Code	Description
C2620000	6F00	Unexpected abend
C2620001	6F01	Unable to test TPL form flag
C2620002	6F02	Unable to load TPL type
C2620003	6F03	TPL storage verification failed
C2620004	6F04	Unable to set TPL active
C2620005	6F05	Unable to update TPL function code
C2620006	6F06	Unable to set TPL 31-bit mode flag off
C2620007	6F07	Unable to follow chain to APCB
C2620008	6F08	Unable to set TPL 31-bit mode flag on
C2620009	6F09	Unable to copy TPL to SAW
C262000A	6F0A	TCEP storage verification failed
C262000B	6F0B	SEPM storage verification failed
C262000C	6F0C	Unable to clear internal ECB
C262000D	6F0D	External ECB storage verification failed
C262000E	6F0E	ADBUF read verification failed
C262000F	6F0F	ADBUF write verification failed
C2620010	6F10	DABUF read verification failed
C2620011	6F11	DABUF write verification failed
C2620012	6F12	Unable to follow MBUF chain
C2620013	6F13	OPBUF read verification failed
C2620014	6F14	OPBUF write verification failed
C2620020	6F20	Function code not positive
C2620021	6F21	Function code greater than max allowed
C2620022	6F22	TPL type not recognized
C2620023	6F23	TPL is already active
C2620024	6F24	AOPEN in 31-bit mode; this function in 24-bit mode

Four-Byte Code	Two-Byte Code	Description
C2620025	6F25	TCEP address is zero
C2620026	6F26	TCEP tag check failed
C2620027	6F27	TCEP pointer to SEPM is zero
C2620028	6F28	SEPM tag check failed
C2620029	6F29	SEPM pointer to TCEP is invalid
C262002A	6F2A	Exit/ECB address is missing
C262002B	6F2B	AOPEN in 24-bit mode; exit/ECB in 31-bit mode
C262002C	6F2C	ECB/exit not on halfword boundary
C262002D	6F2D	ECB not on fullword boundary
C262002E	6F2E	Function table entry invalid for ADBUF
C262002F	6F2F	Function table entry invalid for DABUF
C2620030	6F30	Function table entry invalid for OPBUF
C2620031	6F31	Function not allowed in drain mode
C2620032	6F32	Function not allowed in stop mode
C2620033	6F33	Function not allowed in term mode
C2620034	6F34	ALET of one invalid for ADALT
C2620035	6F35	Invalid ALET specified for ADALT
C2620036	6F36	Invalid ALET specified for DAALT
C2620037	6F37	Invalid ALET specified for DAALT
C2620038	6F38	Invalid ALET specified for OPALT
C2620039	6F39	Invalid ALET specified for OPALT

## C301xxxx (24xx) T01XACPT

The following table lists X'C301' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3010001	2401	Invalid SAW ID
C3010002	2402	Invalid function code
C3010003	2403	Invalid SEPM ID
C3010004	2404	Socket type is not STREAM (invalid socket type)
C3010005	2405	Socket state is not LISTENING (invalid socket state)
C3010007	2407	Call to T01XCREA failed
C3010020	2420	SEPMASEP queue is not empty for non-blocking call
C3010022	2422	SEPMASAW queue is blocked
C3010024	2424	Unicenter TCPaccess is terminating

## C302xxxx (25xx) T01XBIND

The following table lists X'C302' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3020004	2504	SEPM state is not ground for stream socket
C3020005	2505	Address family is not AF_INET
C3020006	2506	Call to T01XINTA failed
C3020007	2507	UDPBIND exit failed
C3020008	2508	TCPBIND exit failed
C3020009	2509	Use of IP address rejected by IPRULE
C302002C	252C	Maximum number of dynamic VIPAs already active

## C303xxxx (26xx) T01XCLS1

The following table lists X'C303' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3030007	2607	T01XCLS1
C3030008	2608	T01XCLS1
C3030009	2609	T01XCLS1
C3030010	2610	T01XCLS1
C3030011	2611	GETSHRS AVXLTCH latch call failed
C3030012	2612	FREESHR SAVXLTCH latch call failed

## C304xxxx (27xx) T01XCLS2

The following table lists X'C304' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3040001	2701	Invalid SAW option code

## C305xxxx (28xx) T01XCONN

The following table lists X'C305' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3050001	2801	Invalid socket type
C3050002	2802	Unable to get SAVXNETL latch
C3050003	2803	Unable to get SAVXNETL latch
C3050009	2809	Unable To free SAVXNETL latch
C305000A	280A	Unable to free SAVXNETL latch
C3050010	2810	Call to T01SUBND failed
C3050030	2830	Address family is not AF_INET for UDP socket
C3050032	2832	Invalid port=0 specified for UDP socket
C3050033	2833	Invalid host=broadcast specified for UDP socket
C3050034	2834	Remote host name not found for UDP socket
C3050050	2850	Invalid state for stream socket
C3050051	2851	Invalid state for stream socket
C3050052	2852	Invalid state for stream socket
C3050053	2853	Invalid address family for stream socket
C3050054	2854	Specified port=0 is invalid for stream socket
C3050055	2855	Invalid host=broadcast specified for stream socket
C3050056	2856	Host name not found for stream socket
C3050057	2857	Non-blocking i/o specified for connect
C3050058	2858	Not enough storage to allocate SAW

## C306xxxx (29xx) T01XUBND

The following table lists X'C306' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3060002	2902	Invalid SAW block ID
C3060003	2903	Invalid SAW function
C3060004	2904	GetSAVXTLAT latch failed
C3060005	2905	Invalid SEPM address
C3060006	2906	Invalid SEPM block ID
C3060007	2907	Free SAVXTLAT latch failed

## C307xxxx (2Axx) T01XCREA

The following table lists X'C307' instance codes.

Four-Byte Code	Two Byte Code	Description
C3070015	2A15	Unable to obtain SAW
C3070016	2A17	Unable to obtain ATCB
C3070017	2A17	Unable to locate SPCB

## C308xxxx (2Bxx) T01XFREE

The following table lists X'C308' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3080001	2B01	ICVT address invalid
C3080002	2B02	SSIT address invalid
C3080007	2B07	Invalid SEPM block ID
C3080008	2B08	Invalid API type
C3080009	2B09	Invalid ASID
C3080010	2B10	Invalid ASCB
C3080011	2B11	Invalid TCB
C3080012	2B12	Allocate latch failed
C3080013	2B13	Invalid ASID
C3080014	2B14	Get SAVXLTCH failed
C3080020	2B20	Unable to obtain SAW

## C30Cxxxx (2Dxx) T01XLIST

The following table lists X'C30C' instance codes.

Four-Byte Code	Two-Byte Code	Description
C30C0001	2D01	Invalid socket type
C30C0002	2D02	Invalid socket state



## C30Dxxxx (2Exx) T01XOPT1

The following table lists X'C30D' instance codes.

Four-Byte Code	Two-Byte Code	Description
C30D0001	2E01	Invalid SAW block ID
C30D0002	2E02	Invalid SAW function code
C30D0003	2E03	Invalid SAW option code
C30D0004	2E04	Invalid option level
C30D0005	2E05	Invalid option level
C30D0006	2E06	Invalid option number
C30D0007	2E07	Option number not supported
C30D0008	2E08	Invalid SAWODATL
C30D0009	2E09	Invalid SAWODATL
C30D000A	2E10	Invalid SAWODATL
C30D000B	2E11	Invalid SAWODATL
C30D000C	2E12	Invalid SAWODATL
C30D000D	2E13	Invalid SAWODATA
C30D000E	2E14	Invalid SAWODATA
C30D000F	2E15	Invalid SAWODATA
C30D0010	2E16	Invalid socket type
C30D0011	2E17	Invalid protocol for SO_SNDLOWAT, must be TCP
C30D0012	2E18	Invalid protocol for SO_RCVLOWAT, must be TCP
C30D0013	2E19	Invalid SAWODATA address
C30D0014	2E20	Send low water mark > maximum size
C30D0015	2E21	Recv low water mark > maximum size
C30D0016	2E22	Invalid MVS release level
C30D0017	2E23	Invalid MVS release level
C30D0018	2E24	Cannot find interface number
C30D0019	2E25	Metric is too big for RIP

Four-Byte Code	Two-Byte Code	Description
C30D001A	2E26	No LNI is up on this interface
C30D001B	2E27	Invalid SAWIOCMD option
C30D001C	2E28	Invalid protocol for TCP_KEEPALIVE, must be TCP
C30D001D	2E29	Invalid protocol for CHECKSUM option, must be UDP
C30D001E	2E30	Invalid protocol for NODELAY option, must be TCP
C30D001F	2E31	Invalid protocol for HDRINCL option, must be RAW
C30D0020	2E32	Invalid protocol for MAXSEGooption, must be TCP
C30D0021	2E33	Metric too big for RIP
C30D0022	2E34	Not enough storage to hold all route table entries
C30D0023	2E35	Invalid option value
C30D0024	2E36	SO_RCVBUF < low water mark
C30D0025	2E37	SO_RCVBUF > maximum size
C30D0026	2E38	TLI_MAXLSEND value is invalid
C30D0027	2E39	TLI_MAXLSEND value < SEPMSLWM
C30D0028	2E40	TLI_MAXLSEND value > XXXXMLS
C30D0029	2E41	TLI_MAXLRECV value is invalid
C30D002A	2E42	TLI_MAXLRECV value < SEPMRLWM
C30D002B	2E43	TLI_MAXLRECV value > XXXXMLR
C30D002C	2E44	TLI_MAXQSEND value is invalid
C30D002D	2E45	TLI_MAXQSEND value < SEPMSLWM
C30D002E	2E46	TLI_MAXQSEND value > XXXXMQS
C30D002F	2E47	TLI_MAXQRECV value is invalid
C30D0030	2E48	TLI_MAXQRECV value < SEPMRLWM
C30D0031	2E49	TLI_MAXQRECV value > XXXXMQR

## C30Exxxx (2Fxx) T01XPEER

The following table lists X'C30E' instance codes.

Four-Byte Code	Two-Byte Code	Description
C30E0001	2F01	Invalid SAW block ID
C30E0002	2F02	Invalid SEPM block ID
C30E0003	2F03	Invalid SAW function code
C30E0004	2F04	Invalid socket state
C30E0005	2F05	Call to T01XINTA failed

## C30Fxxxx (30xx) T01XPRTA

The following table lists X'C30F' instance codes.

Four-Byte Code	Two-Byte Code	Description
C30F0001	3001	Bad protocol; neither TCP nor UDP.
C30F0002	3002	Bad port number; (on free) greater than 65535.
C30F0003	3003	Bad port number; (on use) greater than 65535.
C30F0004	3004	Attempt to free a port that is not in use.
C30F0005	3005	Attempt to use a port already in use.
C30F0006	3006	No ports available; all in use.
C30F0007	3007	The port number passed in the request was out-of-range for the PORTASGN and PORTUSE definitions.
C30F0008	3008	Use of port rejected by PORTRULE.

## C311xxxx (31xx) T01XREAD

The following table lists X'C311' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3110001	3101	Invalid SAW function specified
C3110002	3102	Full block receive exceeds window size
C3110003	3103	Invalid stream socket state
C3110004	3104	Connection reset by peer
C3110005	3105	Connection reset
C3110006	3106	Socket already has uncompleted RECV request (will block)
C3110007	3107	Out of band data not in storage
C3110008	3108	Position not at out of band mark
C3110009	3109	No data or less data than low water mark level
C3110010	3110	Invalid datagram socket state
C3110011	3111	UDP socket requested OOB data (invalid)
C3110012	3112	Non blocking specified, but requests will block
C3110013	3113	No data at end point

## C312xxx T01XOPT2

The following table lists X'C312' instance codes.

Four-Byte Code	Description
C3120001	SAWFUNC is not XOPTION2
C3120002	SAWOPT is not IOCTL or SETSOCKOPT
C3120003	setsockopt level is invalid
C3120004	Option ID is invalid
C3120005	Option length is invalid
C3120006	IP_OPTIONS length exceeds 44
C3120007	IP options lengths exceed total
C3120008	IOCTL bad length
C312000C	SO_DONTROUTE and streams socket
C312000D	IP_TOS value too big
C312000E	SO_BROADCAST and streams socket
C312000F	TCP_NODELAY not streams socket
C3120010	TCP_MAXSEG not streams socket
C3120011	TCP_KEEPALIVE not streams socket
C3120012	UDP_CHECKSUM not datagram socket
C3120013	IP_HDRINCL not raw IP socket
C3120015	TCP_MAXSEG value too big
C3120016	TCP_KEEPALIVE value too big
C3120017	IP_TTL value too big
C3120018	TCP_KEEPALIVE not streams socket
C3120019	IOCTL invalid option
C312001B	TCP_MAXSEG invalid value
C312001C	TCP_KEEPALIVE value is negative
C312001E	IOCTL no MBUF
C312001F	IOCTL cannot find interface number
C3120020	IOCTL add route error

Four-Byte Code	Description
C3120021	IOCTL delete route not found
C3120022	IOCTL delete route error
C3120023	IP timestamp option invalid pointer
C3120024	IP timestamp option length exceeds pointer
C3120025	IP timestamp option length not multiple of four
C3120026	IP timestamp option length not multiple of eight
C3120027	IP source route option invalid pointer
C3120028	IP source route option length too short
C3120029	IP source route option length-3 not multiple of four
C312002A	IP_OPTIONS length too long
C312002B	IOCTL add route already exists
C312002C	Element <i>dvr_version</i> of structure dvreq is invalid
C312002D	Element <i>dvr_length</i> of structure dvreq is invalid
C312002E	Element <i>dvr_option</i> of structure dvreq is invalid
C312002F	IP address already active as a dynamic VIPA
C3120030	IP address is not eligible for application dynamic VIPA
C3120031	Maximum number of dynamic VIPAs already active
C3120032	IP address already active as a standard interface
C3120033	IP address is not active as a dynamic VIPA

## C315xxxx (33xx) T01XSHT1

The following table lists X'C315' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3150001	3301	Invalid SAW function specified
C3150002	3302	SHUT2 ENQ failed, queue is blocked
C3150003	3303	Invalid shutdown option specified
C3150004	3304	Invalid socket state for shutdown 'how'=RECV
C3150005	3305	Invalid socket state for shutdown 'how'=SEND
C3150006	3306	Abortive shutdown by T01XSHT1

## C316xxxx (34xx) T01XSHT2

The following table lists X'C316' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3160001	3401	Set timer function failed

## C317xxx (35xx) T01XSND1

The following table lists X'C317' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3170001	3501	Invalid SAW function code
C3170003	3503	Do not route data invalid for stream socket
C3170004	3504	Receiving only invalid for stream socket
C3170005	3505	Invalid socket state
C3170006	3506	Socket is in reset or disconnect
C3170007	3507	SEPM is invalidated by TCP
C3170009	3509	Cannot block for non blocking request (TCP)
C3170010	3510	Cannot block for non blocking request (UDP)
C3170011	3511	Cannot block for non blocking request, small buffer(UDP)
C3170012	3512	Cannot block for non blocking request, small buffer(UDP)
C3170013	3513	Out of band data invalid for UDP/RAW socket
C3170014	3514	Host address is not provided in SAW (UDP)
C3170015	3515	UDP/RAW socket did not specified port number
C3170016	3516	UDP socket specified SADDR when already connected
C3170017	3517	Data length exceeds maximum packet size
C3170020	3520	Data length exceeds maximum
C3170021	3521	Socket is being shut down
C3170022	3522	Allocate SAW failed
C3170023	3523	Failed to obtain local port and bind it with UDP



## C318xxxx (36xx) T01XSND2

The following table lists X'C318' instance codes.

Four-Byte Code	Two-Byte Code	Description
C3180002	3602	Invalid SEPM state
C3180003	3603	SEPM is invalidated by TCP
C3180004	3604	No mbufs provided
C3180005	3605	SEPM is being shut down

## C404xxxx (3Bxx) T01ASWDN

The following table lists X'C404' instance codes.

Four-Byte Code	Two-Byte Code	Description
C4040001	3B01	Send mbuf failed

## C4FFxxxx (37xx) T01ASFRR

The following table lists X'C4FF' instance codes.

Four-Byte Code	Two-Byte Code	Description
C4FF0001	3701	Function abnormally terminated

## C502xxxxT01ESCF

The following table lists X'C502' instance codes.

Four-Byte Code	Description
C5020001	Socket in FAILED state

## C506xxxxT01ESDI

The following table lists X'C506' instance codes.

Four-Byte Code	Description
C5060001	SEND and RECEIVE queue empty, no data pending
C5060002	SAW indicated it was in WAIT state

## C508xxxxT01ESTP

The following table lists X'C508' instance codes.

Four-Byte Code	Description
C5080001	Shutdown detected

## C512xxxxT01EUCF

The following table lists X'C512' instance codes.

Four-Byte Code	Description
C5120001	Socket in FAILED state

## C516xxxxT01EUDI

The following table lists X'C516' instance codes.

Four-Byte Code	Description
C5160001	SEND and RECEIVE queue empty
C5160002	O/E disconnect received

## C518xxxxT01EUTP

The following table lists X'C518' instance codes.

Four-byte	Description
C5180001	Shutdown detected

## C526xxxx (43xx) T01ETDI

The following table lists X'C526' instance codes.

Four-Byte Code	Two-Byte Code	Description
C5260001	4301	Disconnect received; no send/receive pending
C5260002	4302	Disconnect received; send/receive cancelled
C5260003	4303	Disconnect received; pending select cancelled

## C528xxxx (45xx) T01ETTP

The following table lists X'C528' instance codes.

Four-Byte Code	Two-Byte Code	Description
C5280000	4500	TPEND for pending accept
C5280001	4501	TPEND for CLOSE PASS/OPEN OLD
C5280004	4504	System is stopping
C5280008	4508	System is terminating

## C602xxxx T01SMOVE

The following table lists X'C602' instance codes.

Four-Byte Code	Description
C6020001	No SAW control block provided
C6020002	Invalid SAW control block
C6020003	Not Send/Recv type function
C6020004	No MBUF control block provided
C6020005	Recv buffer is store protected
C6020006	Recv buffer is unavailable
C6020007	Send buffer is fetch protected
C6020008	Send buffer is unavailable
C6020009	Indirect MBUF not allowed for Send
C60200FF	ABEND during data movement

## C704xxxx T01AMIUC

The following table lists X'C704' instance codes.

Four-Byte Code	Description
C7040100	IUCV Declare Buffer failed
C7040200	IUCV Callback function failed
C7040300	IUCV Retrieve Buffer failed
C7040400	IUCV Callback function failed

## C805xxxx T01XTTCP

The following table lists X'C805' instance codes.

Four-Byte Code	Description
C805001	TCP address space has been shut down with a P CLEAR

## C901xxxx T01SIMUX

The following table lists X'C901' instance codes.

Four-Byte Code	Description
C9010001	IP header length exceeds datagram length
C9010002	IP checksum validation failed
C9010003	IP Host unreachable
C9010004	IP protocol not supported
C9010005	UDP port not found
C9010006	TCP session not found
C9010007	IP bad source route option
C9010008	IP header too small
C901000A	IP no MBUFs
C901000B	IP reassembly length error (>64K)
C901000C	IP logic error in reassembly
C901000D	IP logic error in MBUF extent list
C901000E	IP broadcast message not ours
C901000F	IP version is not four
C9010011	MBUF emptied by trimming
C9010012	SEPM queue is blocked

## C909xxx T01SISND

The following table lists X'C909' instance codes.

Four-Byte Code	Description
C9090000	IP Normal discard of original broadcast datagram
C9090001	IP No available LNI could be found
C9090002	IP MBUF does not contain LNI segment
C9090003	IP header will not fit in MBUF
C9090004	IP option with length zero
C9090005	IP options will not fit in MBUF
C9090006	IP options will not fit in MBUF
C9090007	IP bad source route list
C9090008	IP header and options will not fit in MBUF
C9090009	IP option with length zero
C909000A	IP no MBUFs
C909000B	IP broadcast request but SO_BROADCAST not set
C909000C	Unable to find a route because SO_DONTROUTE was set.

## C920xxx T01SUBN

The following table lists X'C920' instance codes.

Four-Byte Code	Description
C9200001	Ilatch get call failed
C9200002	Ilatch free call failed

## C921xxx T01SUIN

The following table lists X'C921' instance code.

Four-Byte Code	Description
C9210001	UDP datagram host/port does not match SEPM
C9210002	Ilatch get call failed
C9210003	UDP datagram length error
C9210004	UDP datagram checksum error
C9210005	UDP datagram receive rejected by user exit
C9210006	UDP error from T01ASCON
C9210007	UDP receive SEPM queue blocked

## C922xxx T01SUSND

The following table lists X'C922' instance codes.

Four-Byte Code	Description
C9220003	UDP failed to get route
C9220004	UDP no MBUFs
C9220007	UDP header does not fit in MBUF
C9220008	UDP datagram send rejected by user exit

## C923xxxx T01SUUNB

The following table lists X'C923' instance codes.

Four-Byte Code	Description
C9230001	Ilatch get call failed
C9230002	Ilatch free call failed

## C930xxxx T01SRBND

The following table lists X'C930' instance codes.

Four-Byte Code	Description
C9300001	Ilatch get call failed
C9300002	Ilatch free call failed
C9300003	RAW bind rejected by user exit

## C931xxx T01SRIN

The following table lists X'C931' instance codes.

Four-byte	Description
C9310001	Ilatch get call failed
C9310002	Ilatch free call failed
C9310003	RAW datagram rejected by user exit

## C932xxxx (5Bxx) T01SRSND

The following table lists X'C932' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9320003	5B03	Host unreachable—route failed.
C9320004	5B04	Resource failure—could not obtain an MBUF.
C9320007	5B07	Request failed by user send exit.



## C933xxxx (5Cxx) T01SRUNB

The following table lists X'C933' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9330001	5C01	ILATCH GET failed.
C9330002	5C02	ILATCH FREE failed.
C9330003	5C03	SEPM not in transport provider table. ( <b>Note:</b> Dead code)

## C940xxxx (5Dxx) T01STCLO

The following table lists X'C940' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9400003	5D03	ILATCH GET failed for TCP Table Latch SAVXTLAT.
C9400004	5D04	ILATCH GET failed for SEPM latch.
C9400005	5D05	ILATCH FREE failed for TCP Table Latch SAVXTLAT.
C9400006	5D06	ILATCH FREE failed for SEPM latch.
C9400007	5D07	ILATCH DEALLOC failed for SEPM latch.
C9400008	5D08	Trace point for endpoint destroy.

## C941xxxx (5Exx) T01STCON

The following table lists X'C941' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9410001	5E01	Wrong state for Connect.
C9410002	5E02	Unable to assign an interface based on local address.
C9410003	5E03	Unable to assign an interface based on local address.
C9410004	5E04	Address combination in use, EADDRINUSE.
C9410005	5E05	Unable to route.
C9410006	5E06	Error return code from Itime service.
C9410007	5E07	ILATCH GET failed for TCP Table Latch SAVXTLAT.
C9410008	5E08	Error return code from Itime service.
C9410009	5E09	ILATCH FREE failed for TCP Table Latch SAVXTLAT.
C941000A	5E0A	Trace point for user connect() call. (TCONNECT)
C941000B	5E0B	Trace point for state change.

## C942xxxx (5Fxx) T01STFIN

The following table lists X'C942' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9420001	5F01	Trace point for user shutdown() request. (TRELEASE)
C9420002	5F02	Trace point for state change.

## C943xxxx (60xx) T01STIMU

The following table lists X'C943' Instance Codes

Four-Byte Code	Two-Byte Code	Description
C9430001	6001	Error return code from Itime service.
C9430002	6002	Trace point for timer update.

## C944xxxx (61xx) T01STIN

The following table lists X'C944' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9440001	6101	Error return code from Itime service.
C9440002	6102	ILATCH GET failed for SEPM latch.
C9440003	6103	ILATCH GET failed for SEPM latch.
C9440005	6105	ILATCH GET failed for TCP Table Latch SAVXTLAT.
C9440006	6106	ILATCH FREE failed for TCP Table Latch SAVXTLAT.
C9440007	6107	ILATCH FREE failed for TCP Table Latch SAVXTLAT.
C9440008	6108	ILATCH GETSHR failed for TCP Table Latch SAVXTLAT.
C9450009	6109	ILATCH FREESHAR failed for TCP Table Latch SAVXTLAT.
C945000B	610B	ILATCH FREE failed for SEPM latch.
C945000C	610C	ILATCH FREE failed for SEPM latch.
C9450014	6114	Trace point for Input.
C9450015	6115	Trace point for Input.
C9450016	6116	Trace point for Input.
C9450017	6117	Trace point for state change.

## C945xxxx (62xx) T01STLIS

The following table lists X'C945' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9450001	6201	Wrong state for Listen.
C9450002	6202	Address combination in use, EADDRINUSE.
C9450003	6203	ILATCH GET failed for TCP Table Latch SAVXTLAT.
C9450004	6204	ILATCH FREE failed for TCP Table Latch SAVXTLAT.
C9450005	6205	Trace point for listen() call. (TLISTEN)
C9450006	6206	Trace point for state change.

## C946xxxx (63xx) T01STREA

The following table lists X'946' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9460001	6301	Duplicate data discarded.
C9460002	6302	Duplicate data discarded.
C9460003	6303	Endpoint cannot receive more.
C9460004	6304	Trimmed TCP header.
C9460005	6305	Trace point for TCP reassembly.

## C947xxxx (64xx) T01STRST

The following table lists X'947' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9470001	6401	Unable to route.
C9470002	6402	Unable to get an MBUF. (Storage shortage)
C9470003	6403	Trace point for output.

## C948xxxx (65xx) T01STSND

The following table lists X'948' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9480001	6501	Send called without an MBUF. (Logic error)
C9480002	6502	Unable to route.
C9480003	6503	Half-open connection and unable to route. (May be SYN flood attack)
C9480004	6504	Error return code from Itime service.
C9480007	6507	Trace point for output.

## C94Cxxxx (69xx) T01STTMK

The following table lists X'C94C' instance codes.

Four-Byte Code	Two-Byte Code	Description
C94C0001	6901	TCP Keepalives exceeded.
C94C0002	6902	Trace point for Keepalive timer.
C94C0003	6903	Trace point for Keepalive output.
C94C0004	6904	Trace point for state change.
C94C0005	6905	Trace point for state change.
C94C0006	6906	Keepalive expired and network unreachable.
C94C0007	6907	Keepalive expired and host unreachable.
C94C0008	6908	Keepalive expired and protocol unreachable.
C94C0009	6909	Keepalive expired and port unreachable.
C94C000A	690A	Keepalive expired and cannot fragment.
C94C000B	690B	Keepalive expired and source route failed.

## C94Dxxxx (6Axx) T01STTML

The following table lists X'C94D' instance codes.

Four-Byte Code	Two-Byte Code	Description
C94D0001	6A01	TCP Linger timer expired.
C94D0002	6A02	Trace point for TCP Linger timer.
C94D0003	6A03	Trace point for state change.
C94D0004	6A04	Linger expired and network unreachable.
C94D0005	6A05	Linger expired and host unreachable.
C94D0006	6A06	Linger expired and protocol unreachable.
C94D0007	6A07	Linger expired and port unreachable.
C94D0008	6A08	Linger expired and cannot fragment.
C94D0009	6A09	Linger expired and source route failed.

## C94Exxxx (6Bxx) T01STTMP

The following table lists X'C94E' instance codes.

Four-Byte Code	Two-Byte Code	Description
C94E0002	6B02	Trace point for TCP Persist timer.

## C94Fxxxx (6Cxx) T01STTMR

The following table lists X'C94F' instance codes.

Four-Byte Code	Two-Byte Code	Description
C94F0001	6C01	TCP Retransmits exceeded.
C94F0002	6C02	Trace point for TCP Retransmit timer.
C94F0003	6C03	Trace point for state change.
C94F0004	6C04	Trace point for state change.
C94F0005	6C05	Retransmits exceeded and network unreachable.
C94F0006	6C06	Retransmits exceeded and host unreachable.
C94F0007	6C07	Retransmits exceeded and protocol unreachable.
C94F0008	6C08	Retransmits exceeded and port unreachable.
C94F0009	6C09	Retransmits exceeded and cannot fragment.
C94F000A	6C0A	Retransmits exceeded and source route failed.



## C950xxxx (6Dxx) T01STMT

The following table lists X'C950' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9500001	6D01	TCP TimeWait state expired.
C9500002	6D02	Trace point for TCP TimeWait timer.
C9500003	6D03	Trace point for state change.
C9500004	6D04	TimeWait expired and network unreachable.
C9500005	6D05	TimeWait expired and host unreachable.
C9500006	6D06	TimeWait expired and protocol unreachable.
C9500007	6D07	TimeWait expired and port unreachable.
C9500008	6D08	TimeWait expired and cannot fragment.
C9500009	6D09	TimeWait expired and source route failed.

## C951xxxx (6Exx) T01STUNB

The following table lists X'C95' instance codes.

Four-Byte Code	Two-Byte Code	Description
C9510003	6E03	ILATCH GET failed for TCP Table Latch SAVXTLAT.
C9510004	6E04	ILATCH FREE failed for TCP Table Latch SAVXTLAT.

## Ilatch Errors

For errors encountered during execution of the ILATCH program, an SVCDUMP is generated.

Return codes are as follows:

GET failur		FREE failur	
x'04'	latch already held.	x'04'	latch not held.
x'08'	logic error.	x'08'	logic error.
x'0C'	bad parameter.	x'0C'	bad parameter.
x'10'	user held local.		

## DGMAKE/AHAP Log Errors

The following table lists DGMAKE/ AHAP log errors.

Code	Description
x'04'	Invalid PID.
x'08'	Protocol is inactive/stopping.
x'0C'	Memory shortage.
x'10'	Invalid PTA address or bad ssb length.
x'14'	Invalid gateway address, or lack of resources.
x'18'	Memory shortage (AHAP get).
x'1C'	Protocol Module Failed.
x'20'	Gethost call failed with bad IP address.
x'24'	Gethost failed with gateway, 0 = network.
x'28'	Gethost failed with gateway, 0 = any network.
x'2C'	Gethost call failed with undefined gateway.
x'30'	Gethost failed with undefined gateway on network.
x'34'	Should not occur.
x'38'	Gethost call failed with insufficient memory.
x'3C'	Address in use (AHAP Log only).

## Security Access Errors

Security Access errors are interpreted as having the last two bytes of the security return code converted to binary; these last two bytes are then interpreted from high to low order as follows:

A BBBB CCCC DDDD

A High order bit on indicates security access error.

BBBBB SAF service call return code.

CCCCC Security return code.

DDDDD Security reason code.

For example, if the return code is C203A104, use the following formula to convert the value of A104 to binary:

A104 = 1010 0001 0000 0100

Then use five bits to determine the error code:

1010000100000100  
rc=8 rc=8 rsn=4

Return codes from SAF service calls are documented in the IBM reference *RACF Callable Services*. GC23-3737-01.

Refer to your system security software documentation for specific security return code descriptions, if you are using a security package other than RACF.



# C Socket errno Codes

This chapter defines errno codes returned by the Unicenter TCPaccess IUCV C socket functions.

When an IUCV C socket function completes, the errno value provides additional information about the error reported. errno values are set when the return code is negative.

## errno Code Table

The values of errno shown in the following table are returned to C socket library applications when the return value is -1.

errno Name	Value	Description
EPERM	1	Not owner
ENOENT	2	No such file
ESRCH	3	No such process
EINTR	4	Interrupted system
EIO	5	I/O error
ENXIO	6	No such device
E2BIG	7	Argument list too long
ENOEXEC	8	Exec format error
EBADF	9	Bad file number
ECHILD	10	No children
EAGAIN	11	No more processes
ENOMEM	12	Not enough core
EACCES	13	Permission denied
EFAULT	14	Bad address

errno Name	Value	Description
ENOTBLK	15	Block device required
EBUSY	16	Mount device busy
EEXIST	17	File exists
EXDEV	18	Cross-device link
ENODEV	19	No such device
ENOTDIR	20	Not a directory
EISDIR	21	Is a directory
EINVAL	22	Invalid argument
ENFILE	23	File table overflow
EMFILE	24	Too many open file
ENOTTY	25	Not a typewriter
ETXTBSY	26	Text file busy
EFBIG	27	File too large
ENOSPC	28	No space left on
ESPIPE	29	Illegal seek
EROFS	30	Read-only file system
EMLINK	31	Too many links
EPIPE	32	Broken pipe
EWouldBlock	35	Operation would block
EINPROGRESS	36	Operation now in progress
EALREADY	37	Operation already in progress
ENOTSOCK	38	Socket operation on
EDESTADDRREQ	39	Destination address required
EMSGSIZE	40	Message too long
EPROTO	41	Protocol wrong type
ENOPROTOPT	42	Protocol not available
EPFNOSUPPORT	43	Protocol not supported
ESOCKTNOSUPPORT	44	Socket type not supported
EOPNOTSUPP	45	Operation not supported
EPFNOSUPPORT	46	Protocol family not supported

errno Name	Value	Description
EAFNOSUPPORT	47	Address family not supported
EADDRINUSE	48	Address already in use
EADDRNOTAVAIL	49	Can't assign requested address
ENETDOWN	50	Network is down
ENETUNREACH	51	Network is unreachable
ENETRESET	52	Network dropped connection
ECONNABORTED	53	Software caused connection
ECONNRESET	54	Connection reset by peer
ENOBUFS	55	No buffer space available
EISCONN	56	Socket is already connected
ENOTCONN	57	Socket is not connected
ESHUTDOWN	58	Can't send after shutdown
ETOOMANYREFS	59	Too many references
ETIMEDOUT	60	Connection timed out
ECONNREFUSED	61	Connection refused
ELOOP	62	Too many levels of nesting
ENAMETOOLONG	63	File name too long
EHOSTDOWN	64	Host is down
EHOSTUNREACH	65	No route to host
ENOTEMPTY	66	Directory not empty
EPROCLIM	67	Too many processes
EUSERS	68	Too many users
EDQUOT	69	Disc quota exceeded
ESTALE	70	Stale NFS file handle
EREMOTE	71	Too many levels of remote in the path
ENOSTR	72	Device is not a stream
ETIME	73	Timer expired
ENOSR	74	Out of streams resources
ENOMSG	75	No message
EBADMSG	76	Trying to read unreadable message

errno Name	Value	Description
EIDRM	77	Identifier removed
EDEADLK	78	Deadlock condition
ENOLCK	79	No record locks available
ENONET	80	Machine is not on network
ERREMOTE	81	Object is remote
ENOLINK	82	The link has been severed
EADV	83	ADVERTISE error
ESRMNT	84	SRMOUNT error
ECOMM	85	Communication error
EPROTO	86	Protocol error
EMULTIHOP	87	Multihop attempted
EDOTDOT	88	Cross mount point
EREMCHG	89	Remote address change
EIBMBADCALL	1000	A bad socket-call constant was found in the IUCV header
EIBMBADPARM	1001	Other IUCV header error, bad length, and so forth
EIBMSOCKOUTOFRANGE	1002	Socket number assigned by client interface code (for socket() and accept()) is out of range
EIBMSOCKINUSE	1003	Socket number assigned by client interface code is already in use
EIBMIUCVERR	1004	Request failed due to IUCV error
EIBMCONFLICT	1008	Conflicting call already outstanding on socket
EIBMCANCELLED	1009	Request cancelled via socket function CANCEL request



## IPRCODE Table

The codes shown in this table are returned to applications by the IUCV transport and may be displayed as C socket or macro API return codes.

IPRCODE	Description
1	Invalid path ID
2	Path quiesced – no sends allowed
3	Message limit exceeded
4	Priority messages not allowed on path
5	Buffer too short for message
6	Fetch protection exception
7	Addressing exception
8	Message ID found, but class/path invalid
9	Message has been purged
10	Message length negative
11	Target is not logged on
12	Target has not declared a buffer
13	Invoker max connections exceeded
14	Target maximum connections exceeded
15	Not authorized to connect to target
16	Invalid CP system service name
17	Invalid function code
18	Invalid message limit
19	Already has declared a buffer
20	Path has been severed
21	Parameter list message not allowed
22	Send list invalid
23	Negative length in list
24	Reserved: VM/SP uses this IPRCODE to flag invalid total list length
25	PRMMSG and BUF/ANSLIST not allowed
26	Buffer list not double-word aligned

IPRCODE	Description
27	Answer list not double-word aligned
28	No control buffer exists

# DNR Return Codes

This chapter describes all the return codes generated on completion (whether normal or abnormal) of the DIRSRV macro instruction.

It includes:

- [General Return Codes](#) – Describes general return codes that indicate the general success or failure of a DIRSRV macro instruction
- [Recovery Action Codes](#) – Describes recovery action codes that define categories of errors that can be handled in a common manner
- [Conditional Completion Codes: RTNCD](#) – Describes conditional completion codes that indicate unusual conditions that accompany an otherwise normal completion of the function request
- [Specific Error Codes: xxxx](#) – Describes specific error codes that provide more definition of a failure
- [Exceptional Conditions: RTNCD 04xx](#) – Describes possible exceptional conditions
- [Execution Environment Errors: RTNCD 08xx](#) – Describes execution environment errors
- [Format or Specification Errors: RTNCD 0Cxx](#) – Describes format or specification errors
- [Sequence and Procedural Errors: RTNCD 10xx](#) – Describes sequence and procedural errors
- [Logic Errors with No DPL Return Code: RTNCD 14xx](#) – Describes logic errors that do not return a DPL return code

The DIRSRV macro instruction completes normally (or conditionally) when the requested information is returned in the storage area provided by the application program. The length of the storage area is updated to reflect the actual amount of information returned.

On normal return to the application program, the general return code in register 15 is set to zero (DROKAY), and the conditional completion code is returned in register zero. If the macro instruction was executed in synchronous mode, the general return code indicates whether the directory service completed normally. If the macro instruction was executed in asynchronous mode, the general return code indicates whether the directory service request was accepted.

If the DIRSRV macro instruction completes abnormally, no information is returned in the storage area and the storage area length is unmodified. The general return code in register 15 and the recovery action code indicate the nature of the failure. If the general return code is set to DEFAILED, the recovery action code is returned in register zero and the DPL return code contains a specific error code that identifies a particular error. If the general return code is set to DEFATLPTL, the recovery action code and the error code are both returned in register zero and the DPL is not updated.

## General Return Codes

General return codes are returned in register 15 and indicate the general success or failure of a DIRSRV macro instruction. If the macro instruction was executed in synchronous mode, the general return code indicates whether the directory service completed normally. If the macro instruction was executed in asynchronous mode, the general return code indicates whether the directory service request was accepted.

The general return codes defined in this section are those generated by the DNR.

### 0 (X'00') DROKAY

Reason: Request accepted, or completed normally or conditionally.

If the macro instruction was executed in synchronous mode, the function completed normally (or conditionally) without any errors. Register 0 contains a conditional completion code indicating what, if any, unusual conditions occurred. If the macro instruction was executed in asynchronous mode, the request was accepted and the directory request was initiated.

### 4 (X'04') DRFAILED

Reason: Request not accepted, or request completed abnormally due to a non-recoverable error or abnormal condition.

If the macro instruction was executed in synchronous mode, the function completed abnormally. If the macro instruction was executed in asynchronous mode, the request was not accepted because of an error or abnormal condition.

Register 0 contains a recovery action code.

**8 (X'08') DRFATLPL**

Reason: Request not accepted, or completed abnormally because of a fatal DPL error.

A fatal error occurred, preventing the normal completion or initiation of a function request. Errors in this category are typically caused by an invalid DPL address or a corrupted DPL. Register 0 contains a recovery action code and a specific error code identifying the specific error condition.

## Recovery Action Codes

The recovery action code is stored in the DPL return code field (DPLRTNCD) and returned in register zero whenever the general return code in register 15 indicates a failure. The recovery action code defines categories of errors that can be handled in a common manner.

If the macro instruction was executed in synchronous mode, the recovery action code is returned to the application program in register zero. If the macro instruction was executed in asynchronous mode and the request was not accepted, the recovery action code is returned in register zero. If the request was accepted and subsequently failed processing, the recovery action code is returned in the DPL.

**0 (X'00') DROKAY**

Reason: Request accepted, or completed normally or conditionally.

If the macro instruction was executed in synchronous mode, DAOKAY returned in register zero indicating the request completed normally with no conditionals.

DAOKAY returned in register zero indicating the request was accepted. On completion of the asynchronous request, DAOKAY stored in the recovery action field of the DPL indicates normal or conditional completion with the conditional completion code (if any) returned in the specific error code field of the DPL.

**4 (X'04') DAEXCPTN**

Reason: An exceptional condition occurred that prevented normal completion or acceptance of the directory function.

### **8 (X'08') DAENVIRO**

Reason: Failure due to abnormal environmental condition.

This code typically indicates an abnormal condition in the execution environment that is outside the direct control of the application program.

Some external action may be required to relieve the condition.

### **12 (X'0C') DAFORMAT**

Reason: Format or specification error.

A failure occurred because of a format or specification error, usually associated with a parameter provided by the application program in a DIRSRV request.

### **16 (X'10') DAPROCED**

Reason: Sequence or procedural error.

The failure occurred because the DPL was already in use from a previous DPL-based request. This error typically indicates a logic error in the application program and should not occur once the program is debugged.

The recovery action code and the specific error code are returned to the application program in register zero.

### **20 (X'14') DADPLERR**

Reason: Logic error with no DPL return code.

A logic error occurred, but the DPL associated with the request was in a state or condition that prevents storing the recovery action code and error code in the return field of the DPL.

The recovery action code and the specific error code are returned to the application program in register zero.

## Conditional Completion Codes: RTNCD 00xx

Conditional completion codes are returned in register zero whenever the general return code in register 15 indicates normal or conditional completion. The conditional completion code is also stored in the DPL return code field (DPLRTNCD) in place of the specific error code (DPLERRCD). The recovery action code is set to zero (DAOKAY) to indicate normal or conditional completion.

Conditional completion codes are used to indicate unusual conditions that accompany an otherwise normal completion of the function request. These codes typically indicate the occurrence of a condition that normally does not affect the successful execution of the application program. They should not be treated as an error.

Each bit of the eight-bit completion code represents a particular condition. Therefore, unlike specific error codes, a conditional completion code can represent the presence of more than one condition.

### 0 (X'00') DROKAY

Reason: Normal completion with no conditionals.

### 08 (X'08') DCLOCAL

Reason: Local configuration data was used to resolve a global request.

### 16 (X'10') DCNAMEIA

Reason: The character string used in a GET-HOST-BYNAME request was an Internet address in dot notation (that is, 127.0.0.1). The value buffer contains the four-byte Internet address in hexadecimal (that is, x'7F000001').

### 32 (X'20') DCOVRFLO

Reason: The storage area identified by the qualified name buffer was not large enough to return the fully qualified domain name associated with the return information.

### 64 (X'40') DCALIAS

Reason: The host name used for the request was an alias. If a qualified name buffer was specified, the fully qualified name was returned in the storage area.

**128 (X'80') DCMORE**

Reason: The return information is a list of values and there were more entries than would fit into the storage area or the number of entries exceeded the size parameter.

## Specific Error Codes: xxxx

Specific error codes provide a more definitive characterization of the failure, and can be used in combination with the recovery action code to determine the precise error recovery procedure.

### Valid Combinations

There are multiple valid combinations of recovery action codes and specific error codes. All valid combinations of return code settings are shown in the following table.

**Note:** An entry in the table indicates that the column and row values are valid combinations for DPLRTNCD. The label appearing in the table is the mnemonic defined in the DPL DSECT.

Value	DAEXCPTN X'04'	DAENVIRO X'08'	DAFORMAT X'0C'	DAPROCED X'10'	DADPLERR X'14'
X'00'					
X'01'	DENONAME	DESYSERR	DEBDOPCD	DEACTIVE	DEBDTYPE
X'02'	DENOVALU	DESUBSYS	DEBDFNCD		DEPROTCT
X'03'	DENOQNAM	DENOTCNF	DEBDXECB		DEPLMODE
X'04'	DETIMOUT	DENOTACT	DEBDEXIT		
X'05'	DERFAIL	DENOTRDY	DEBDNAME		
X'06'	DENOTFND	DESTOP	DEBDVALU		
X'07'	DENOCDS	DEUNAVBL	DEBDQNAM		
X'08'	DENAMERR	DERSOURC			
X'09'	DEOVRFLO	DENOTPRB			
X'0A'	DENOBLOK	DETERM			
X'0B'	DENODATA				



Value	DAEXCPTN X'04'	DAENVIRO X'08'	DAFORMAT X'0C'	DAPROCED X'10'	DADPLERR X'14'
X'0C'	DENAMODE				
X'0D'	DEVAMODE				
X'0E'	DEQNMODE				

## Exceptional Conditions: RTNCD 04xx

This section lists exceptional conditions.

### 1 (X'01') DENONAME

Reason: Name buffer specification exception.

The NABUF field is specified as zero or the length of the storage area is specified as zero in NALEN.

The following DNR DENONAME diagnostic codes apply:

Diagnostic Code	Description
X'0400'	Field NABUF was zero.
X'0404'	Field NALEN was zero.

### 2 (X'02') DENOVALU

Reason: Value buffer specification exception.

The VABUF field is specified as zero or the length of the storage area is specified as zero in VALEN.

The following DNR DENOVALU diagnostic codes apply:

Diagnostic Code	Description
X'0400'	Field VABUF was zero.
X'0404'	Field VALEN was zero.

### 3 (X'03') DENOQNAM

Reason: Qualified name buffer specification exception. The QNBUF field is specified as nonzero and the length of the storage area is specified as zero in QNLEN.

The following DNR DENOQNAM diagnostic code applies:

Diagnostic Code	Description
X'0400'	Field QNLEN was zero.

### 4 (X'04') DETIMOUT

Reason: A time limit was specified (TIME) as nonzero and the limit was exceeded. A time value of zero specifies no limit.

The following DNR DETIMOUT diagnostic codes apply:

Diagnostic Code	Description
X'0800'	The specified time limit has been exceeded before the request could be satisfied.
X'0E07'	RECVFR request timed out.
X'0F07'	SENDTO request timed out.

### 5 (X'05') DERFAIL

Reason: Resolver failed. An unrecoverable error occurred during processing a Domain Name System (DNS) response.

The following DNR DERFAIL diagnostic code applies:

Diagnostic Code	Description
X'0000'	The DNR was processing a Domain Name System (DNS) response and a limit to the number of alias referrals (four) was exceeded.

## 6 (X'06') DENOTFND

Reason:

Answer not found. The directory information was not found. If the request involved a local search, the information was not found in the local configuration members. If the search involved a global search of the Domain Name System (DNS), the retry transmission limits specified in the DNR configuration file (DNRCFGxx) were exceeded.

The following DNR DENOTFND diagnostic codes apply:

Diagnostic Code	Description
X'0000'	The search string was not found in a local configuration file.
X'0001'	DNR could not resolve the DIRSRV request with local data and cannot use a name server because NAMESERVER(NONE) was specified in DNRCFGxx.
X'0010'	Domain name server request failed. OPTCD=LOCAL was specified.
X'0011'	Domain name server request failed. OPTCD=GLOBAL was specified. The DNR was configured for local operation.
X'0601'	The retransmission limit specified in the MAXSENDS parameter in the DNR configuration member (DNRCFGxx) was exceeded.
X'0603'	The transmission limit for the name server list specified in the CYCLEMAX parameter in the DNR configuration member (DNRCFGxx) was exceeded.

## 7 (X'07') DENOCDS

Reason: A DNR configuration member required for enabling the DNR to satisfy the request was empty.

The following DNR DENOCDS diagnostic codes apply:

Diagnostic Code	Description
X'0200'	A GET-HOST-BYNAME request received by the DNR and the DNR host name configuration member specified in the DNR configuration member was empty (HOSTTABLE(DNRHSTxx)). If OPTCD=GLOBAL, the name server configuration member (NAMESERVER(DNRNSCxx)) is also empty.
X'0300'	A GET-HOST-BYVALUE request received by the DNR and the DNR host name configuration member specified in the DNR configuration member was empty HOSTTABLE(DNRHSTxx)). If OPTCD=GLOBAL, the name server configuration member (NAMESERVER(DNRNSCxx)) was also empty.
X'0400'	A GET-HOST-BYALIAS request with OPTCD=LOCAL received by the DNR and the DNR alias configuration member specified in the DNR configuration member was empty ALIAS(DNRALCxx)). If OPTCD=GLOBAL, the name server configuration member (NAMESERVER(DNRNSCxx)) was also empty.
X'0500'	A GET-NETWORK-BYNAME request received by the DNR and the DNR network configuration member specified in the DNR configuration member was empty (NETWORK(DNRNETXX)).
X'0600'	A GET-NETWORK-BYVALUE request received by the DNR and the DNR network configuration member specified in the DNR configuration member was empty (NETWORK(DNRNETXX)).
X'0700'	A GET-SERVICE-BYNAME request received by the DNR and the DNR services configuration member specified in the DNR configuration member was empty (SERVICES(DNRSVCxx)).
X'0800'	A GET-SERVICE-BYVALUE request received by the DNR and the DNR services configuration member specified in the DNR configuration member was empty (SERVICES(DNRSVCxx)).

Diagnostic Code	Description
X'0900'	A GET-PROTOCOL-BYNAME request received by the DNR and the DNR protocol configuration member specified in the DNR configuration member was empty (PROTOCOL(DNRPRT <sub>xx</sub> )).
X'0A00'	A GET-PROTOCOL-BYVALUE request received by the DNR and the DNR protocol configuration member specified in the DNR configuration member was empty (PROTOCOL(DNRPRT <sub>xx</sub> )).
X'0B00'	A GET-HOSTSERV-BYNAME, GET-HOSTINFO-BYNAME or GET-ROUTE-BYNAME request received by the DNR and the DNR name server configuration member specified in the DNR configuration member was empty (NAMESERVER(DNRNSC <sub>xx</sub> )).

## 8 (X'08') DENAMERR

Reason:

Host does not exist. The search string is syntactically correct, but the host is not identified in the Domain Name System (DNS) namespace. If the search string was found in the local alias configuration member (DNRALC<sub>xx</sub>), the incorrect search string is returned in the storage area identified by DPLQNBUFF. Otherwise, the search string is the string identified in the DPLNABUF operand.

The following DNR DENAMERR diagnostic codes apply:

Diagnostic Code	Description
X'0000'	The search string was the fully qualified name specified in the NABUF operand.
X'0001'	The search string was a fully qualified replacement string found in the alias configuration file for the alias specified in the NABUF operand.
X'0002'	The search strings were the result of concatenating the search list strings to the partial name specified in the NABUF operand.
X'0003'	The search strings were the result of concatenating the search list strings to the partial name found in the alias configuration file for the alias specified in the NABUF operand.
X'0004'	The search strings were the result of concatenating the search list strings to the partial name found in the alias configuration file for the alias specified in the NABUF operand, but were not found.

**9 (X'09') DEOVRFLO**

Reason: Reply data exceeds data area length. The return data exceeded the return storage area. For requests that return a list of responses, a single response could not be returned.

The following DNR DEOVRFLO diagnostic codes apply:

<b>Diagnostic Code</b>	<b>Description</b>
X'0000'	Return data exceeded return storage area.
X'0400'	Module DNRRCSRB determined that the name buffer was not large enough to receive the returned name. This condition should have been determined by an earlier module but did not indicate a possible internal logic error.
X'0404'	Module DNRRCSRB determined that the value buffer was not large enough to receive the returned name. This condition should have been determined by an earlier module but did not indicate a possible internal logic error.
X'0408'	Module DNRRCSRB determined that the qualified name buffer was not large enough to receive the returned name. This condition should have been determined by an earlier module but did not indicate a possible internal logic error.

**10 (X'0A') DENOBLOK**

Reason: Request not completed because OPTCD=NOBLOCK was specified.

The request was received specifying OPTCD=NOBLOCK and the data was not readily available in the DNR cache containing previously answered requests.

The following DNR DENOBLOK diagnostic code applies:

<b>Diagnostic Code</b>	<b>Description</b>
X'0000'	The data was not readily available.

## 11 (X'0B') DENODATA

Reason: Insufficient data available to satisfy request.

The search string was syntactically correct and the string identified a host that exists in the Domain Name System (DNS) name space but there is no configuration data to satisfy this type of request.

The following DNR DENODATA diagnostic codes apply:

Diagnostic Code	Description
X'0000'	The name server did not have the specific resource record configured for the host name.
X'0001'	<p>A name server has responded to a DNS query. It responds with a NO error response but without any information to resolve the request.</p> <p>The name server may not be configured or may be misconfigured for the node.</p> <p>If the name server responding is not listed in the DNRNSCxx configuration file when running in NONRECURSIVE mode (see the RECURSIVE   NONRECURSIVE parameter on the DNR statement in file DNRCFGxx), there may be other name servers responding to DNS requests that have not been configured for the current host issue. They can respond for the current domain while your local name servers are having problems responding to requests in a timely fashion.</p> <p><b>Note:</b> Consider setting the RECURSIVE parameter on the DNR statement in file DNRCFGxx.</p>

## 12 (X'0C') DENAMODE

Reason: Name buffer addressing conflict.

The address of the name buffer does not match the addressing mode of the application program issuing the DIRSRV request. The name buffer must be below the 16 MB line if the addressing mode of the program is 24 (AMODE=24).

The following DNR DENAMODE diagnostic codes apply:

Diagnostic Code	Description
X'0400'	The application was executing with AMODE=24 and bits 0-7 of the name buffer address were not zero.

### 13 (X'0D') DEVAMODE

Reason: Value buffer addressing conflict.

The address of the value buffer does not match the addressing mode of the application program issuing the DIRSRV request. The value buffer must be below the 16 MB line if the addressing mode of the program is 24 (AMODE=24).

The following DNR DEVAMODE diagnostic codes apply:

Diagnostic Code	Description
X'0400'	Application was executing with AMODE=24 and bits zero-seven of the value buffer address were not zero.

### 14 (X'0E') DEQMODE

Reason: Qualified name buffer addressing conflict.

The address of the qualified name buffer does not match the addressing mode of the application program issuing the DIRSRV request. The qualified name buffer must be below the 16 MB line if the addressing mode of the program is 24 (AMODE=24).

The following DNR DEQMODE diagnostic codes apply:

Diagnostic Code	Description
X'0400'	Application was executing with AMODE=24 and bits zero-seven of the qualified name buffer address were not zero.
X'0104'	An ABEND occurred in module DNRRCSRB, which executes as a MVS SRB to complete a directory request. This error is accompanied by a software record written to SYS1.LOGREC.
X'0400'	An ABEND occurred in module DNRPDREQ while running the RB chain of the currently dispatched task (TCB). This error should only occur if MVS internal RB chaining is severely modified.
X'0404'	An ABEND occurred in module DNRCREQ, which executes as a cross memory PC routine to move a directory request from the application's address space to the DNR address space. This error is accompanied by a dump if a local dump data set (that is, SYSUDUMP) was preallocated and if the ABEND normally produces a dump. This error is also accompanied by a software record written to SYS1.LOGREC.



Diagnostic Code	Description
X'0408'	Module DNRPDREQ, which executes in the same mode as the calling application to verify and initialize a directory request, was unable to successfully establish an ESTAE.
X'040C'	<p>An ABEND occurred in module DNRPDREQ, which executes in the same mode as the calling application to verify and initialize a directory request. The ABEND was not expected (some S0C4 abends are expected) and the directory request is abnormally terminated.</p> <p>This error is accompanied by a dump if a local dump data set (that is, SYSUDUMP) was preallocated by the requesting application and if the ABEND normally produces a dump.</p> <p>This error is also accompanied by a software record that is written to SYS1.LOGREC.</p>
X'0410'	<p>An ABEND occurred in module DNRCSSIX, which executes as a cross memory PC routine, to mark a directory request purged when a DIRSRV PURGE macro is executed.</p> <p>This error is accompanied by a dump if a local dump data set (that is, SYSUDUMP) was preallocated by the requesting application and if the ABEND normally produces a dump.</p> <p>This error is also accompanied by a software record that is written to SYS1.LOGREC.</p>
X'0800'	An ABEND occurred in the DNR address space processing this directory request.

Diagnostic codes X'11xx' through X'2002' indicate that an internal logic error was detected in one of the DNR completion routines.

This error was triggered by either:

- The completion routine being passed a null resource record pointer or a null or negative resource record count (diagnostic codes X'xx01') D or D
- The resources record count exceeding the number of valid resource records (diagnostic codes X'xx02')

The first byte of the diagnostic code identifies the routine that determined the error. This error is accompanied by a DNR153E error message in the SYSPRINT log of the Unicenter TCPaccess job.

The following table lists DNR DESYSERR error routines.

<b>Diagnostic Code</b>	<b>Routine Determining Error</b>
X'1101'	NRTDA (GET HOST BY NAME)
X'1102'	NRTDA (GET HOST BY NAME)
X'1201'	NRTDPTR (GET HOST BY VALUE)
X'1301'	NRTDCNA (GET HOST BY ALIAS)
X'1A01'	NRTDWKS (GET HOSTSERVICES BY NAME)
X'1A02'	NRTDWKS (GET HOSTSERVICES BY NAME)
X'1B01'	NRTDINF (GET HOSTINFO BY NAME)
X'1C01'	NRTDMX (GET ROUTE BY NAME)
X'1C02'	NRTDMX (GET ROUTE BY NAME)
X'1F01'	NRTDNS (GET NS BY NAME)
X'1F02'	NRTDNS (GET NS BY NAME)
X'2001'	NRTDSOA (GET SOA BY NAME)

## Execution Environment Errors: RTNCD 08xx

This section lists execution environment errors.

### 1 (X'01') DESYSERR

Reason: MVS system error occurred. An MVS system ABEND or error was detected trying to process the directory request or return reply.

The following diagnostic codes apply:

<b>Diagnostic Code</b>	<b>Routine Determining Error</b>
X'0100'	An ABEND occurred in module DNRCDCMP, which executes as a cross memory PC routine to move completion data from the DNR address space to the address space requesting the directory service. This error is accompanied by a software record written to SYS1.LOGREC.
X'0104'	An ABEND occurred in module DNRRCSRB, which executes as a MVS SRB to complete a directory request. This error is accompanied by a software record written to SYS1.LOGREC.

<b>Diagnostic Code</b>	<b>Routine Determining Error</b>
X'0400'	An ABEND occurred in module DNRPDREQ while running the RB chain of the currently dispatching task (TCB). This error should only occur if MVS internal RB chaining is severely modified.
X'0404'	An ABEND occurred in module DNRCDDREQ, which executes as a cross memory PC routine to move a directory request from the application's address space to the DNR address space. This error is accompanied by a dump, if a local dump data set (that is, SYSUDUMP) was preallocated and if the ABEND normally produces a dump. This error is accompanied by a software record written to SYS1.LOGREC.
X'0408'	Module DNRPDREQ, which executes in the same mode as the calling application to verify and initialize a directory request, was unable to successfully establish an ESTAE.
X'040C'	An ABEND occurred in module DNRPDREQ, which executes in the same mode as the calling application to verify and initialize a directory request. The ABEND was not expected (some SOC4 ABENDs are expected) and the directory request is abnormally terminated. This error is accompanied by a dump if a local dump data set (that is, SYSUDUMP) was preallocated by the requesting application and if the ABEND normally produces a dump. This error is accompanied by a software record written to SYS1.LOGREC.
X'0410'	An ABEND occurred in module DNRCSSIX, which executes as a cross memory PC routine, to mark a directory request purged when a DIRSRV PURGE macro is executed. This error is accompanied by a dump if a local dump data set (that is, SYSUDUMP) was preallocated by the requesting application and if the ABEND normally produces a dump. This error is also accompanied by a software record that is written to SYS1.LOGREC.
X'0800'	An ABEND occurred in the DNR address space processing this directory request.

Diagnostic codes X'11xx' through X'2002' indicate that internal logic error was detected in one of the DNR completion routines.

This error was triggered by either:

- The completion routine being passed a null resource record pointer or a null or negative resource record count (diagnostic codes X'xx01') D or D
- The resources record count exceeding the number of valid resource records (diagnostic codes X'xx02')

The first byte of the diagnostic code identifies the routine that determined the error. This error is accompanied by a DNR153E error message in the SYSPRINT log of the Unicenter TCPaccess job.

The following diagnostic codes apply:

<b>Diagnostic Code</b>	<b>Routine Determining Error</b>
X'1101'	NRTDA (GET HOST BY NAME)
X'1102'	NRTDA (GET HOST BY NAME)
X'1201'	NRTDPTR (GET HOST BY VALUE)
X'1301'	NRTDCNA (GET HOST BY ALIAS)
X'1A01'	NRTDWKS (GET HOSTSERVICES BY NAME)
X'1A02'	NRTDWKS (GET HOSTSERVICES BY NAME)
X'1B01'	NRTDINF (GET HOSTINFO BY NAME)
X'1C01'	NRTDMX (GET ROUTE BY NAME)
X'1C02'	NRTDMX (GET ROUTE BY NAME)
X'1F01'	NRTDNS (GET NS BY NAME)
X'1F02'	NRTDNS (GET NS BY NAME)
X'2001'	NRTDSOA (GET SOA BY NAME)

## 2 (X'02') DESUBSYS

Reason:

MVS subsystem error. DNR operates as a MVS subsystem and maintains subsystem control blocks that identify items such as whether DNR is available, common program entry points and cross memory PC numbers.

The following DNR DESUBSYS diagnostic codes apply:

Diagnostic Code	Description
X'0400'	<p>On entry to module DNRPDREQ, which verifies and initiates directory requests, register zero must point to the DNR Network Directory Services Anchor (NDSA) block. Normal expansion of the DIRSRV macro causes register zero to be loaded with the NDSA address prior to branching into DNRPDREQ.</p> <p>Verify that the expansion of DIRSRV is valid, that DNR initialized or terminated normally, or if calling DNRPDREQ without using the DIRSRV macro, that register zero is loaded with the address of the NDSA.</p>
X'0404'	<p>On entry to module DNRPDREQ, which verifies and initiates directory requests, register zero must point to the DNR Network Directory Services Anchor (NDSA) block. While trying to access the NDSA, a fetch protection occurred. Normal expansion of the DIRSRV macro causes register zero to be loaded with the NDSA address prior to branching into DNRPDREQ.</p> <p>Verify that</p> <ul style="list-style-type: none"> <li>■ The expansion of DIRSRV is valid</li> <li>■ DNR initialized or terminated normally</li> <li>■ If calling DNRPDREQ without using the DIRSRV macro, that register zero is loaded with the address of the NDSA</li> </ul>
X'0408'	<p>On entry to module DNRPDREQ, which initiates DIRSRV PURGE requests, register zero must point to the DNR Network Directory Services Anchor (NDSA) block. Normal expansion of the DIRSRV macro causes register zero to be loaded with the NDSA address prior to branching into DNRPDREQ.</p> <p>Verify that:</p> <ul style="list-style-type: none"> <li>■ The expansion of DIRSRV is valid</li> <li>■ DNR initialized or terminated normally</li> <li>■ If calling DNRPDREQ without using the DIRSRV macro, that register zero is loaded with the address of the NDSA</li> </ul>

### 3 (X'03') DENOTCNF

Reason: Subsystem not configured in MVS. DNR operates as a MVS subsystem and maintains subsystem control blocks that identify items such as whether DNR is available, common program entry points and cross memory PC numbers.

This diagnostic DNR DENOTCHF code applies:

Diagnostic Code	Description
X'0000'	<p>The subsystem ID specified in the SYSID= parameter of the DIRSRV macro (field DPLSYSID of the DPL) could not be located on the MVS subsystem chain.</p> <p>Verify that:</p> <ul style="list-style-type: none"><li>■ DNR is active</li><li>■ The specified subsystem ID is the same</li></ul> <p><b>Note:</b> The subsystem ID is case sensitive. Generally all subsystem ID's are in upper case.</p>

### 4 (X'04') DENOTACT

Reason: Subsystem not started or active. DNR operates as a MVS subsystem and maintains subsystem control blocks that identify items such as whether DNR is available, common program entry points and cross memory PC numbers.

The following DNR DENOTACT diagnostic code applies:

Diagnostic Code	Description
X'0000'	<p>The subsystem control blocks owned by DNR indicate that DNR has not started or is not currently active.</p>

## 5 (X'05') DENOTRDY

Reason: Subsystem is not fully initialized. Try later. DNR operates as a MVS subsystem and maintains subsystem control blocks that identify items such as whether DNR is available, common program entry points and cross memory PC numbers.

The following DNR DENOTRDY diagnostic codes apply:

Diagnostic Code	Description
X'0000'	The DNR subsystem control blocks indicate that the address space is active but the DNR has not yet initialized.
X'0400'	The Network Directory Services Anchor (NDSA) address was set in the subsystem control blocks but it is marked not yet ready to accept directory requests.

## 6 (X'06') DESTOP

Reason: Subsystem is in the process of shutting down. If the DNR operator request DNR to stop, then requests in progress are terminated and new requests are refused.

The following DNR DESTOP diagnostic codes apply:

Diagnostic Code	Description
X'0400'	The Network Directory Services Anchor (NDSA) block indicates that the DNR is stopping because of an operator request and this directory request was not accepted.
X'0800'	The directory request was being processed but before it could be completed, an operator requested stop of the DNR task group was issued.

## 7 (X'07') DEUNAVBL

Reason: Unrecoverable transport error (API error). The DNR uses the Unicenter TCPAccess API to communicate with Domain Name Servers. An unrecoverable error occurred on at least one transport request.

The following DNR DEUNAVBL diagnostic codes apply:

Diagnostic Code	Description
X'0801'	Memory allocation error when trying to establish a network connection.
X'0802'	Error in issuing an AOPEN.

Diagnostic Code	Description
X'0803'	Error in issuing a TOPEN.
X'0B04'	Error in issuing a TINFO.
X'0B05'	Error in issuing a TOPTION.
X'0B06'	Memory allocation error when trying to establish a network connection.
X'0B07'	Error in issuing a TBIND.
X'4xyy'	The DNR received an error issuing a TCHECK call on a TRECEIVE TPL. The TPL recovery action code is designated by the <i>x</i> . The specific error code is designated by the <i>yy</i> .
X'8xyy'	The DNR received an error issuing a TCHECK call on a TSEND TPL. The TPL recovery action code is designated by the <i>x</i> . The specific error code is designated by the <i>yy</i> .

## 8 (X'08') DERSOURC

Reason: There were insufficient resources to successfully process the directory services request.

The following DNR DERSOURC diagnostic codes apply:

Diagnostic Code	Description
X'xxyy'	An integer representing the function is designated by the <i>x</i> . An integer representing the specific line number is designated by the <i>yy</i> .
X'0014'	Module DNRCDDREQ returned with RC=16, indicating that no XWA was available.
X'0404'	Module DNRCDDREQ returned with an unexpected return code: RC=04.
X'0408'	Module DNRCDDREQ returned with RC=08 indicating that there was insufficient storage available to allocate a DSRB in the DNR address space.
X'0410'	Module DNRCDDREQ returned with an unexpected return code: RC=20.
X'040C'	Module DNRCDDREQ returned with RC=12 indicating that there was insufficient storage available to allocate buffers needed for this request in the DNR address space.



**9 (X'09') DENOTPRB**

Reason: Task issuing request was not running as a Program Request Block (PRB)

The DNR requires that directory requests are only to be executed from applications running as a MVS PRB. Applications running as a SRB or IRB can not issue directory requests.

The following DNR DENOTPRB diagnostic code applies:

<b>Diagnostic Code</b>	<b>Description</b>
X'0400'	The application program was not executing as an MVS PRB.

**10 (X'0A') DETERM**

Reason:

DNR task group has terminated.

DNR has a subsystem task termination exit that detects when tasks related to the issuance or processing of directory requests have terminated.

The following DNR DETERM diagnostic code applies:

<b>Diagnostic Code</b>	<b>Description</b>
X'0800'	The subsystem task termination exit detected that the DNR task within the DNR address space terminated without going through shutdown logic, neither slow nor fast.

## Format or Specification Errors: RTNCD 0Cxx

This section lists format or specification errors.

### 1 (X'01') DEBDOPCD

Reason:

Field DPLOPCTD in the DPL contained invalid bit settings.

Field DPLOPTCD in the DPL consists of bits that identify different processing options the DNR should use while processing this request.

The following DNR DEBDOPCD diagnostic code applies:

Diagnostic Code	Description
X'0400'	A bit that is not supposed to be set was. <ul style="list-style-type: none"><li>■ If using the DIRSRV macro, verify that the storage was cleared to binary zeroes before issue the DIRSRV macro</li><li>■ If calling the DNR without using the DIRSRV macro, verify that only option bits defined in the DPL are actually set before making the call</li></ul>

### 2 (X'02') DEBDFNCD

Reason:

Invalid DPL function code set

Field DPLFNCCD in the DPL must contain a value between 1 (X'01') and 12 (X'0C').

The following DNR DEBDFNCD diagnostic codes apply:

Diagnostic Code	Description
X'0400'	The value of DPLFNCCD exceeds the maximum value, DFMAX that is equal to 12 (X'0C').
X'0404'	Field DPLFNCCD was not set (that is, it equals zero).

**3 (X'03') DEBDXECB**

Reason: ECB=external-ecb was specified on the DIRSRV macro but the external ECB specified in field DPLECB was invalid.

The following DNR DEBDXECB diagnostic codes apply:

Diagnostic Code	Description
X'0400'	<p>Either a fetch or a protection exception occurred while trying to access the external ECB specified.</p> <p>Verify that the address of the external ECB is valid and that the storage protect key of the external ECB matches that of the application program issuing the directory request.</p>
X'0404'	<p>In the DPL, flag DPLFXECB was set but the value in field DPLXECB was zero.</p> <p>If the application is building the DPL itself, make sure a valid ECB address is specified if flag DPLFXECB is set.</p>
X'0408'	<p>An external ECB must be located below the 16 MB line if the addressing mode of the application is 24 (AMODE=24).</p> <p>Verify that bits zero seven of the external ECB address are zero if the addressing mode is 24 or change the addressing mode to 31 (AMODE=31) if the ECB is located above the 16 megabyte line.</p>
X'040C'	<p>An ECB must be located on a full word boundary. The ECB specified was not fullword aligned. Correct the alignment of the ECB.</p>

**4 (X'04') DEBDEXIT**

Reason: The asynchronous exit specified in field DPLEXIT is not valid. EXIT=exit was specified on the DIRSRV macro but the asynchronous exit address specified was invalid.

The following DNR DEBDEXIT diagnostic codes apply:

Diagnostic Code	Description
X'0400'	<p>A protection exception occurred while trying to access the instruction located at the specified exit address.</p> <p>Verify that the address of the asynchronous exit is valid and that the storage where the asynchronous exit exists is not fetch protected.</p>

Diagnostic Code	Description
X'0404'	In the DPL, flag DPLFEXIT was set but the value in field DPLEXIT was zero.  If the application is building the DPL itself, make sure a valid exit address is specified if flag DPLFEXIT is set.
X'0408'	The asynchronous exit must be located below the 16 MB line if the addressing mode of the application is 24 (AMODE=24).  Verify that bits 0-7 of the exit address are zero if the addressing mode is 24 or change the addressing mode to 31 (AMODE=31) if the exit is located above the 16 MB line.
X'040C'	The asynchronous exit must be located on a half word boundary. The exit specified was not fullword aligned. Correct the alignment of the exit or set the correct exit address.

## 5 (X'05') DEBDNAME

Reason: Bad name buffer. The name buffer address is illegal or the name specified in the buffer is syntactically incorrect.

The following DNR DEBDNAME diagnostic codes apply:

Diagnostic Code	Description
X'0400'	A protection exception occurred trying to either store into or reference data in the name buffer: <ul style="list-style-type: none"><li>■ The name buffer must be located in non-fetch protected storage if it is being used as the request field</li><li>■ It must be located in non-store protected storage if is being used as the reply field</li></ul> Verify that the name buffer has the same storage protection key as the application, that it was not accidentally freed and that the address passed in field DPLVABUF was really the address of the value buffer.
X'0103'	The data in the storage area identified by the NABUF operand request does not conform to the syntax rules.
X'0104'	The data in the storage area identified by the NABUF operand found in the local alias configuration member (DNRALCxx). The replacement string for the alias does not conform to the syntax rules.

Diagnostic Code	Description
X'0200'	<p>A NABUF address was passed but the NALEN field was invalid (for example, NALEN=0).</p> <p>You must pass a valid positive length in the NALEN field when setting the NABUF field.</p>

## 6 (X'06') DEBDVALU

Reason:

Bad value buffer.

The value buffer address is illegal or the value specified in the buffer is syntactically incorrect.

The following DNR DEBDVALU diagnostic codes apply:

Diagnostic Code	Description
X'0400'	<p>A protection exception occurred trying to either store into or reference data in the value buffer:</p> <ul style="list-style-type: none"> <li>■ The value buffer must be located in non-fetch protected storage if it is being used as the request field</li> <li>■ It must be located in non-store protected storage if is being used as the reply field</li> </ul> <p>Verify that the value buffer has the same storage protection key as the application, that it was not accidentally freed and that the address passed in field DPLVABUF was really the address of the value buffer.</p>
X'0200'	<p>The data in the storage area identified by the VABUF operand does not conform to the syntax rules.</p>

## 7 (X'07') DEBDQNAM

Reason: Protection exception accessing the qualified name buffer.

The qualified name buffer must be located in non-store protected storage.

The following DNR DEBDQNAM diagnostic code applies:

Diagnostic Code	Description
X'0400'	<p>A protection exception occurred trying to store data into the qualified name buffer.</p> <p>Verify that the:</p> <ul style="list-style-type: none"><li>■ Qualified name buffer has the same storage protection key as the application</li><li>■ It was not accidentally freed</li><li>■ Address passed in field DPLQNBUFF is really the address of the qualified name buffer</li></ul>

## Sequence and Procedural Errors: RTNCD 10xx

This section lists sequence and procedural errors.

### 1 (X'01') DEACTIVE

Reason: The DPL was still in use by a previous DIRSRV request.

A DPL cannot be reused for another directory request until a prior directory request using the same DPL has completed.

The following DNR DEACTIVE diagnostic codes apply:

Diagnostic Code	Description
X'0000'	<p>The DIRSRV macro expansion (with MF=E or MF=M) detected that the DPL was still in use (field DPLACTIV not zero).</p> <ul style="list-style-type: none"><li>■ If the DPL is still in use from a previous asynchronous request, issue the MVS WAIT macro to suspend execution until the previous request completes or use another DPL.</li><li>■ If the DPL is not really in use from a prior request, make sure that the storage obtained for the DPL was cleared to binary zeros before issuing the DIRSRV macro.</li></ul>

Diagnostic Code	Description
X'0400'	Module DNRPDREQ, which verifies directory requests, detected that the DPL was still in use (field DPLACTIV not zero). <ul style="list-style-type: none"><li>■ If the DPL is still in use from a previous asynchronous request, issue the MVS WAIT macro to suspend execution until the previous request completes or use another DPL.</li><li>■ If the DPL is not really in use from a prior request, make sure that the storage obtained for the DPL was cleared to binary zeroes before issuing the DIRSRV macro.</li></ul>

## Logic Errors with No DPL Return Code: RTNCD 14xx

This section lists logic errors with no DPL return code.

### 1 (X'01') DEBDTYPE

Reason: Invalid DPL identification field.

Field DPLIDENT must be initialized to value DPLIDSTD, which is equal to 237 (X'ED') on entry to module DNRPDREQ.

The following DNR DEBDTYPE diagnostic code applies:

Diagnostic Code	Description
X'0400'	Field DPLIDENT was not set to value DPLIDSTD. Verify that the correct form (MF=) of the DIRSRV macro is being used. If building the DPL, make sure that field DPLIDENT is properly initialized.

## 2 (X'02') DEPROTCT

Reason: DPL was fetch or store protected.

The DPL must be located in storage with the same storage protection key as the application and must not be fetch protected.

The following DNR DEPROTCT diagnostic codes apply:

Diagnostic Code	Description
X'0400'	<p>The DPL was located in fetch protected storage.</p> <p>Verify that the address of the DPL being passed to module DNRPDREQ is in fact the address of the DPL.</p> <p>For programs on XA machines, verify that:</p> <ul style="list-style-type: none"><li>■ The DPL is below the 16 MB line if the addressing mode of the application is 24 (AMODE=24)</li><li>■ The storage is owned by the current task or is being shared between tasks</li></ul>
X'0404'	<p>The DPL was located in store protected storage.</p> <p>Verify that the address of the DPL being passed to module DNRPDREQ is in fact the address of the DPL.</p> <ul style="list-style-type: none"><li>■ If running APF authorized, make sure that the PSW storage key is the same now as when the storage was obtained</li><li>■ For programs on XA machines, verify that the DPL is below the 16 MB line if the addressing mode of the application is 24 (AMODE=24)</li></ul>



# FTP Server Messages

---

This chapter explains the causes of the messages issued by the FTP Server and suggests appropriate responses.

## FTP Server Messages

Messages are presented in numerical order.

### 121 Bye noted, will logout when transfer completes

Reason: The user requests logoff from the host, and when the current file transfer completes, the logoff is processed.

### 125 List started OK

Reason: The LIST/NLST/DIR command started.

### 125 Transfer started

Reason: The file transfer is started.

**150 Dataset opened; data connection starting.**

**150-Data transfer type is *type*. Structure is *struct*. Mode is *mode*.**  
**150-Dataset name: *dsname* Dataset attributes: *Dsorg=dsorg Recfm=recfm***  
**150-Lrecl=*lrecl* Blksize=*blksize* Volser=*volser* Unit=*unit***  
**150-Primary allocation is *tracks1* tracks. Secondary allocation is *tracks2***  
**tracks.**  
**150-Network data which exceeds LRECL will be wrapped to the next**  
**record.**

Reason: This is a multi-line server message produced during data transfer. The 150 message provides information about the file being transferred and about the settings associated with the data transfer.

The following table shows the fields in this 150 message:

Field	Description
<i>type</i> <i>struct</i> <i>mode</i>	Describe the settings associated with the data transfer. Refer to the FTP chapters of the <i>User Guide</i> for details about these settings.
<i>dsname</i>	Indicates the data set being transferred.
<i>lrecl</i> <i>blksize</i> <i>volser</i> <i>unit</i> <i>tracks1</i> <i>tracks2</i>	Indicates the value for each field. These fields relate to the data set being transferred in the <i>dsname</i> field.

**200 Note: Ignored, overridden by site space**

Reason: The space allocation information supplied by the user is ignored because the space allocation was previously specified in a SITE command.

**200 OK, Ready**

Reason: FTP is ready for command processing.

**200 Port request OK.**

Reason: The PORT command was accepted.

**200 Site command was accepted**

Reason: The SITE command was accepted.

**200- % Free Free Largest Free**

Reason: This reply is issued by the FTP server in response to the SITE QDISK command. It provides information on DASD devices.

**202 ACCT not needed, ignored**

Reason: The accounting information supplied by the user is unnecessary and is ignored.

**211 ---Status---**

Reason: This message marks the beginning of the listing from the STAT command.

**211 <End of Status>**

Reason: This message marks the end of the listing from the STAT command.

**214 ---HELP---**

Reason: This message marks the beginning of the listing from the HELP command.

**214 <end of HELP>**

Reason: This message marks the end of the listing from the HELP command.

**215 MVS is the operating system of this server**

Reason: This reply is issued by the FTP server in response to the SYST command. It identifies the operating system on which the server resides.

**220 Enter USER command with userid operand**

Reason: This message directs the user to supply a user command and user ID.

### **220 Logged out, parms reset, enter USER command and ID**

Reason: The user ID for logon to either Host A or Host B is logged off the host. The values for BYTE, ALLO, TYPE, STRU, and MODE parameters are reset to the defaults.

### **220 Logged out, parms retained, enter USER command**

Reason: The user ID for logon to either Host A or Host B is logged off the host. The values for BYTE, ALLO, TYPE, STRU, and MODE parameters are saved.

### **220 *sitename* FTP Server, Enter command or HELP**

Reason: The FTP server process at site *sitename* is ready to accept commands.

### **221 Session terminated**

Reason: The command connection either to Host A or to Host B is closed.

### **221 Quit command received. Goodbye.**

Reason: The command connection to either Host A or to Host B is closed.

### **226 Abort command completed**

Reason: The abort command issued successfully aborted the file transfer.

### **226 Empty file transfer complete – 0 (zero) data bytes sent**

Reason: The file transfer completes successfully, and the disk file of the retrieving host is closed. However, the file contains zero data bytes.

### **226 Transfer complete**

Reason: The file transfer completes successfully, and the disk file of the receiving host is closed.

**226- Transfer complete. number bytes sent/received in secs seconds (rate bytes/s)**

Reason: The FTP server produces a multi-line 226 response after a successful data transfer. Within the 226 response, statistics about the data transfer are reported back to the user. The following multi-line 226 response read data from an MVS data set and sent it across the network.

```
Transfer complete.  
 3439 bytes sent in 2.49 seconds (1381 bytes/s)  
Path FILE.NAME User UID Data bytes sent 6480  
Disk tracks read 1  
226
```

The following table lists the fields in this 226 message:

Field	Description
<i>numbers bytes rate</i>	Statistics about bytes transferred across the network and the rate of transfer per second.
<i>dsname</i>	Name of the data set read from disk and transferred over the network.
<i>user</i>	Indicates that user ID UID initiated the file transfer.
<i>numwritten</i>	Total number of data bytes read from disk for dsname.
<i>tracks</i>	Total number of disk tracks read.

If the file transfer had been read from the network and written to a file on MVS, a multi-line 226 response might look like this:

```
226-Transfer complete  
 19 bytes received in 8.00 seconds (2 bytes/s)  
Dataset name: MVS.TEMP.DATA User UID Data bytes received 17  
Disk tracks written 1 Records padded 1  
226
```

This message is similar to the previous multi-line 226 response, except the data transfer is from the network to be stored on an MVS data set.

Depending on the attributes of the file being written to, certain events happen to records as they are placed into a file.

Some messages that might appear in this 226 response are:

Field	Description
Records padded <i>n</i>	Indicates the total number of records that had pad characters inserted at the end to accommodate the file structure.
Records truncated <i>n</i>	Indicates the total number of records truncated because the record sent was larger than the record size for the file.
Records folded <i>n</i>	Indicates the total number of records broken into multiple records because the record sent was larger than the record size for the file.
Records suspect <i>n</i>	Indicates total number of records shipped in block mode by the sending side that may contain suspect data (that is, the data sent is suspected of errors and is not reliable).

### 226 Transfer complete (unique file name: *filename*)

Reason: If SUNIQUE filename was specified, this message reflects the remote host's unique file name.

### 230- Logged in - User=*user* Working directory "*dir*"

Reason: The user *user* is logged in with working directory *dir*. The IP address of the FTP server is address.

### 250 Deleted OK

Reason: The specified file is deleted successfully.

### 250 Renamed OK

Reason: The specified file is renamed successfully.

### 250 File action OK

Reason: The file action is performed successfully.

**250 Data will be written to NULLFILE**

Reason: This reply is issued by the FTP server when a change directory (CD/CWD) command is received with a path name of \*DEV.NULL. Subsequent data transfers that cause data to be written to the file system by the FTP server are written to a dummy data set (NULLFILE). To reverse this, enter another change directory command, specifying a different path name.

**250 dsname deleted.**

Reason: This reply is issued by the FTP server in response to the DELE command. It identifies the data set that was successfully deleted.

**250 Job cancelled OK**

Reason: The specified JES job ID deleted successfully.

**250 List completed successfully.**

Reason: The LIST/NLST/DIR command completed successfully.

**250 Transfer completed successfully.**

Reason: The file transfer completed successfully.

**257 prefix**

Reason: *prefix* is the current default prefix for the user.

**257 No prefix defined**

Reason: There is no default prefix defined.

**257 "'pathname'" partitioned dataset is current directory**

Reason: The partitioned data set '*pathname*' is the working directory.

**257 "'pathname'" partitioned dataset created with attributes:**

Reason: A partitioned data set was created in response to a MKD command. The file attributes follow.

### **331 Enter PASS command**

Reason: The user is directed to enter a password command and password.

### **331 Logged out, parms retained, enter PASS command**

Reason: The user is logged out, and the values for BYTE, ALLO, TYPE, STRU, and MODE parameters are saved.

### **331 Logged out, parms reset, enter PASS command**

Reason: The user is logged out, and the values for BYTE, ALLO, TYPE, STRU, and MODE parameters are not saved.

### **332 Enter ACCT command**

Reason: The user is directed to enter an accounting command and accounting information.

### **350 Requested file action pending further information**

Reason: This is an interim reply indicating the completion of a part of a multi-part request. For instance, a RNFR (ReName FRom) command has been received, and the server is awaiting a RNT0 (ReName TO) command.

### **421 Operator forced logout**

Reason: A privileged user in DDNMVSOP mode (the operator control mode of OPEN-Link for IBM/MVS) cancels the FTP session.

### **422 Host network software error, incomplete**

Reason: FTP processing is incomplete because a software error was encountered in the host network's software.

### **425 Unable to open connection**

Reason: The connection could not be opened.



**426 Data connection closed. Transfer incomplete**

Reason: File transfer is incomplete because the data connection between hosts is lost.

**426 Data transfer aborted**

Reason: The file transfer operation aborts.

**426 Data transfer aborted. Ready**

Reason: The file transfer aborts and FTP is ready for further user commands.

**426 Data transfer timeout, aborted**

Reason: The DATAIDLE time specified for FTP expired before the last receive request completed. The transfer aborts.

**426 Invalid RDW length detected on input file. Transfer incomplete**

Reason: File transfer is incomplete because input file is variable blocked and an invalid record descriptor word (RDW) length value was detected.

**450 All access paths to volume busy**

Reason: No access paths are available to write a file to or read a file from the device (volume). 450 Data set tied up by another user.

Another user has exclusive control of the data set to be processed by FTP. Generally, the other user issued a PUT command to the data set causing an enqueue for exclusive control. Unicenter TCPaccess provides resource serialization on the data set level, which has certain consequences for partitioned data sets. In particular, only one user at a time can access a data set for the purpose of storing data. If one FTP user is attempting to store a member into a partitioned data set, other users are prevented from accessing the same data set, even if the access is for a different member. However, multiple users can simultaneously retrieve members from the same partitioned data set because retrieve operations do not require exclusive control.

**450 No path to volume**

Reason: The operating system cannot access the device (volume) from which a file is to be read or to which a file is to be written.

**450 OBTAIN FAILED FOR *oper* PROCESSING**

Reason: This message is sent when the VTOC information cannot be retrieved.

The type of operation being processed to the PDS (*oper*) can be one of the following:

DELETE	Delete a member.
RENAME	Rename a member.

**450 Output file had errors during I/O processing in previous transfer.**

Reason: The transfer request is rejected because the last transfer attempt was for the same output file and resulted in an error (Msg 451 Transfer aborted. Error during I/O processing)

**450 *function* FUNCTION FAILED RETURN CODE = *nn* QNAME = *qname*  
RNAME = *dsname*[\_*membername*] LEN = *lll* function FAILED FOR *oper* PROCESSING**

Reason: The type of MVS function attempted for this operation (*oper*), where *function* is one of the following:

ENQUEUE	Enqueue a resource to the MVS system.
DEQUEUE	Dequeue a resource from the MVS system.
RESERVE	Reserve the disk volume where the PDS resides.

The return code generated by the above MVS function (*nn*) is in hexadecimal. The return codes for the MVS function are described in the IBM *Application Development Reference* documentation set.

The name of the queue (*queue*) is one of the following:

SPFEDIT	Used for text, source, or object data set.
SYSIEWLP	Used for load module data set.

The DSNAMES of the PDS is represented by (*dsname*). The name of the PDS member to be processed is represented by *membername*; the length of the RNAME field, in decimal, is represented by *lll*. The length of *dsname* is 44; if *membername* is included, the length of the RNAME field is 52.

The type of operation being processed to the PDS (*oper*) is one of the following:

DELETE	Delete a member.
RENAME	Rename member.
STOR	Store a member.

This message is sent when the PDS member is tied up by another user.

#### **451 Aborting transfer, network block header invalid**

Reason: A block mode (Mode B) file transfer was in progress and a block header received contained invalid data in the flag field of the header.

#### **451 Aborting transfer, network block header invalid**

Reason: A block mode (Mode B) file transfer was in progress and a block header received contained invalid data in the flag field of the header. This can be caused by the sender not being in Mode B.

#### **451 Character translation failed, transfer incomplete**

Reason: An attempt to load a single- or double-byte character set translation table failed.

#### **451 Data set cannot be opened**

Reason: The file to be processed by FTP cannot be opened for reading or writing.

#### **451 Host software error**

Reason: A host software error occurs, causing abnormal termination of the requested action.

#### **451 HSM recall wait time expired, request cancelled**

Reason: An HSM recall of a migrated file did not occur within a specified amount of time. The request is cancelled.

Action: Verify system default parameters with system programmer or issue a SITE RECALL command to increase default wait time.

#### **451 I/O error detected in data set**

Reason: The file being read or written contains an I/O error.

#### **451 I/O error in data set, transfer incomplete**

Reason: The file being written at the receiving host or read from the sending host contains an I/O error. The file transfer terminates abnormally.

#### **451 Magnetic tape volume cannot be mounted**

Reason: The requested data set is currently allocated to another user, or cannot be mounted at this time.

Action: Retry the FTP transfer later, or contact the tape librarian or operator for a reason, as appropriate.

#### **451 Network interface module not available**

Reason: One of the modules required to initiate or complete the file transfer was missing at the time of command execution.

#### **451 Open/mount of tape data set failed; rtn=rrrrrrrr**

Reason: An attempt to mount and open a data set on magnetic tape failed. The return code is *rrrrrrrr*.

#### **451 Request cancelled by operator**

Reason: A request to mount a tape was cancelled by the operator. The request is cancelled.

Action: Retry the FTP transfer later, or contact the tape librarian or operator for a reason, as appropriate.

#### **451 Requested magnetic tape unit(s) not available**

Reason: Either all tape units of the type requested are currently off-line or allocated to other users, or a PARALLELMOUNT or UNITCOUNT parameter has requested more units than are currently available.

Action: Retry the FTP transfer when devices are available.

**451 Tape mount wait time expired; request cancelled**

- Reason: The wait time specified in the configuration or on a SITE command for a tape mount has expired. The request is cancelled.
- Action: Retry the FTP transfer later, or contact the tape librarian or operator for a reason, as appropriate.

**451 Transfer aborted. Error during I/O processing. System code is xxx-rc**

- Reason: During either End-of-Volume (EOV) or close processing, the data management DCB ABEND exit was driven. An ABEND would have occurred had the exit not suppressed the ABEND. The file transfer is terminated. The system code of the suppressed ABEND is represented by *xxx*. The reason code of the suppressed ABEND is represented by *rc*.
- Action: Check the appropriate MVS manuals to identify the cause of the ABEND using the ABEND code *xxx* and the reason code *rc*. If the ABEND is issued during EOV processing due to insufficient space in the data set, make sure sufficient space exists in the data set or on the volume before restarting the file transfer. Certain close ABENDs such as the B14 occur when there is insufficient room in the PDS directory or PDS data areas. After correcting the condition causing the ABEND, restart the file transfer.

**451 Transfer aborted. I/O error detected. SYNAD data is jobname, stepname, unit, type, ddname, operation, error, address, BSAM.**

- Reason: During the CHECK of either a READ or WRITE macro, the synchronous error exit was driven. This exit extracted the error data listed in line two of the error via the SYNADF macro. The file transfer was terminated.
- Action: Attempt to diagnose the error using data provided by SYNADF. If possible, correct it and restart the data transfer. If the error is a wrong length record condition, it is likely that a record in the data set exceeds the data set's block size. If this is the case, either delete the record or PDS member or change the attributes of the data set to have a larger block size.

**451 Transfer completed abnormally. Completion code is Sxxx**

- Reason: An ABEND with the specified system ABEND code (xxx) occurred during the file transfer. The file transfer did not complete.
- Action: If the ABEND is due to an I/O error condition such as end of volume or data set close, correct the data management problem that caused the error and retry the file transfer. If the ABEND is of a programmatic nature, contact Customer Support.

**451 Transfer completed abnormally, Completion code is Uxxxx**

- Reason: Either the data transfer PTASK PABENDED with the specified user code (xxx) or the data transfer task PEXITed with the specified return code (xxx). Module FTPSFTRD could not match the code with a list of known exit codes and cannot determine if the file transfer was successful. It is assumed that the file transfer failed.
- Action: Contact Customer Support for assistance in diagnosing the cause of this message.

**451 I/O error while updating PDS directory. Directory is possible full**

- Reason: A CLOSE macro was issued after writing a member of a partitioned data set and an ABEND SB14 occurred indicating an I/O error occurred while updating the PDS directory. This condition is accompanied by MVS system message IEC217 B14-xx, which explains in detail the reason IOS issued the SB14 ABEND. This error usually occurs when there is no room left in a PDS directory.

**451 Transfer incomplete due to system error**

- Reason: A host system error occurred during file transfer. The file transfer halts abnormally before completion.

**451 VTOC full**

- Reason: The table of contents on the volume (VTOC) to which a file is to be written is full and can accept no more entries.

**451 System error in locating data set, R1=xxxxxxx**

Reason: The host system had a problem with the allocation. Other IBM SVC99 messages may follow if SMS is coded on the GLOBAL statement in the APPCFGxx member. These messages are in the IBM documentation, *MVS/ESA Vx System Messages* Volume 1-5, GC28-1656 through GC28-1660. The value xxxxxxxx is the dynamic allocation (SVC99) return code.

**451 Transfer aborted. Transport error detected, RTNCD=XXYYZZZZ. See Messages and Codes Reference for API Return Codes.**

Reason: This message indicates that the FTP transfer was aborted. The XX of RTNCD is the recovery action codes; YY is the specific error code; ZZZZ is the diagnostic and sense codes.

Action: Refer to the API Return Codes for a complete Reason of RTNCD.

**452 No core to execute operation now**

Reason: No memory is available on the host attempting to execute the FTP command.

**452 No core to interpret command now**

Reason: No memory is available on the host attempting to interpret the FTP command.

**500 Command exceeds 1276 characters, ignored**

Reason: The entered command exceeds the maximum length of 1276 characters.

**500 Command unrecognized**

Reason: The FTP software does not recognize the entered command.

**500 Empty line, ignored**

Reason: An empty command line is entered and FTP ignored it.

**500 Excessive operands or ending parenthesis not found**

Reason: Excessive operands found for a valid keyword or a parenthesis is missing from the command line.

### **500 Ending quote not found**

Reason: The ending quote expected at the end of the FTP command was not entered.

### **500 Invalid delimiter syntax**

Reason: Delimiter entered is unrecognizable or invalid.

### **500- Unable to recall DCBDSN *dcbdsn\_name*.**

Reason: A SITE DCBDSN command was received, but the model data set is migrated and cannot be recalled (possibly because NORECALL was specified in the FTP configuration, or by a previous SITE command).

Action: If possible, recall the data set. Otherwise, change the DCBDSN data set name.

### **500- Unable to locate DCBDSN *dcbdsn\_name*.**

Reason: A SITE DCBDSN command was received, but the model data set cannot be found.

Action: Change the DCBDSN data set name to the name of a cataloged data set.

### **500- DCBDSN data set *dcbdsn\_name* is not on DASD.**

Reason: A SITE DCBDSN command was received, but the model data set does not reside on a direct access device.

Action: Change the DCBDSN data set name to the name of a cataloged data set on a DASD device.

### **500- The DCBDSN data set name is too long.**

Reason: A SITE command was received that specified the DCBDSN parameter, but the data set name specified (when appended to the current prefix) exceeds 44 characters.

Action: Check the data set name specified. If it is a fully qualified data set name, enclose it in single quotes.



**500- parameter *parameter* is unrecognized.**

Reason: A SITE command was received that specified an unknown parameter.

Action: See the *User Guide* for a list of SITE parameters and syntax.

**500- parameter *keyword* has an invalid subparameter value.**

Reason: A SITE command was received which contained a keyword parameter specifying an invalid value for one or more subparameters.

Action: See the *User Guide* for a list of SITE parameters and syntax.

**500- EXPDT cannot be specified with RETPD**

Reason: A SITE command was received which contained the EXPDT parameter, but the RETPD was previously specified. EXPDT and RETPD are mutually exclusive. The EXPDT parameter is ignored.

Action: Use SITE RESET to remove the RETPD value before specifying EXPDT.

**500- RETPD cannot be specified with EXPDT**

Reason: A SITE command was received which contained the RETPD parameter, but the EXPDT had been previously specified. EXPDT and RETPD are mutually exclusive. The RETPD parameter is ignored.

Action: Use SITE RESET to remove the EXPDT value before specifying RETPD.

**500- The maximum length of the parameter *parameter* is *length* characters.**

Action: A SITE command was received that contained a keyword parameter specifying an over-long subparameter.

Action: See the *User Guide* for a list of SITE parameters and syntax.

**500- The keyword *keyword* requires at least one subparameter.**

Reason: A SITE command was received containing a keyword parameter that requires a subparameter, but no subparameter was specified.

Action: See the *User Guide* for a list of SITE parameters and syntax.

**500 SITE command was accepted with errors.**

Reason: Indicates that one or more errors were detected in the SITE command.

This message is preceded by one or more 500 replies. Refer to accompanying messages to determine appropriate action.

**501 Invalid or conflicting parameters, command ignored**

Reason: Invalid or conflicting parameters are specified in the command.

**501 Invalid value on RECFM keyword**

Reason: The record format value entered in the SITE RECFM record format is invalid.

**501 MOUNT waittime set to system configured maximum**

Reason: A MOUNT request specified a greater value than the maximum wait time allowed by the system configuration.

**501 Padding value should be Z (zeroes), O (ones), or B (blanks)**

Reason: A char value other than Z, O, or B is specified in the SITE PAD char command.

**501 Parameter value not in correct numerical range**

Reason: A parameter value specified in the command line has a value outside its valid numeric range.

**501 Requested tape label option is not permitted by system.**

Reason: A SITE command specified BLP or NL where that LABEL option is not allowed by the system configuration.

**501 Required operand or keyword value not found**

Reason: A required operand or keyword was omitted from the command.

**501 SITE CHARSET/DECSSET command failed to load table. Command not implemented**

Reason: An attempt to load a single- or double-byte character set translation table failed.

**501 SITE RECALL command specified an invalid value of 0 for HSM wait time**

Reason: An invalid value of zero (0) is specified on the SITE RECALL command.

Action: Reissue SITE RECALL command with a value of 1-1439.

**501 Wildcard characters are not permitted within a partitioned dataset name**

Reason: Wildcard characters (\*) are not permitted within the data set name of a partitioned data set.

**501 Member name not permitted; MVS does not support subdirectories**

Reason: An MKD command was received specifying a directory name within the current PDS directory.

**501 Wildcard characters (\* and %) may only be specified in last qualifier in directory mode**

Reason: A LIST or NLST command was received in directory mode and the path name specified a wildcard pattern character (\* or %) in a qualifier other than the last qualifier. In directory mode, only the final qualifier can contain wildcard characters.

Action: Change the path name or enter data set mode.

**502 Command not implemented**

Reason: The requested facility is not implemented on the host, and the command cannot be performed.

**502 Data set list functions not implemented, use TSO command LISTCAT or LISTDS**

Reason: The FTP functions that list data set information have not been implemented on the host. The user should use the LISTCAT and LISTDS TSO commands.

**502 HSM is not configured to system. Command not implemented**

Reason: A SITE RECALL command is issued, but HSM is not defined to the system.

Action: Verify ACFSFTP macro parameter HSM is not NORECALL. Edit APPCFGxx with HSM defined to FTP.

**502 Tape processing is not configured. Command not implemented.**

Reason: A SITE command was received to modify a parameter for tape, but tape processing was disallowed by the system administrator.

**502 Unimplemented MODE type C, command ignored**

Reason: Mode type compress (C) is not supported by the FTP Server.

**503 Abort ignored, no data transfer in progress**

Reason: The ABORT command is ignored because no file transfer was in progress when it was issued.

**503 APPE/REST not implemented for Magnetic Tape datasets.**

Reason: A RESTART or APPEND was attempted for a data set on magnetic tape. These commands are not supported for magnetic tape.

**503 Command conflicts with previous commands**

Reason: The entered command does not logically follow previous commands and cannot be processed.

**503 Command unexpected after ALLO, ignored**

Reason: The entered command is unacceptable because it is entered after the ALLO function for the specified file.

**503 Command unexpected at this point, refused**

Reason: The entered command is unacceptable at this time and is refused.

**503 Expected RNTD, RNFR ignored**

Reason: The entered RENAME command is invalid.

**503 Expected STOR, APPE, or RETR, REST ignored**

Reason: The RESTART command is ignored because the STORE, RETRIEVE, or APPEND command was expected.

**503 Login required, enter USER**

Reason: The message directs the user to supply user ID information to the host system.

**503 Unable to logout until data transfer completes**

Reason: Logoff from the host cannot be processed until the current file transfer completes.

**503 SUBMIT requires STOR command, ignored**

Reason: The command entered is not acceptable because a PUT command is not accepted after a SITE SUBMIT.

**504 Not implemented for that parameter, ignored**

Reason: A parameter used on the preceding command is not implemented on the specified host system.

**504 Option not implemented**

Reason: The requested processing option is not implemented on the specified host.

**504 Unsupported combination of TYPE and STRU**

Reason: The specified file TYPE/STRUcture combination is not supported by FTP.

**504 Restart requires BLOCK mode**

Reason: BLOCK mode must be specified for a restart.

**520 Network connection open error**

Reason: The CONN command request cannot be completed successfully. The open for the remote host connection fails.

### **520 Storage shortage, causes TOPEN Failure**

Reason: Server FTP cannot create a data connection due to an SOS failure during the TOPEN for the data connection session.

Action: Ensure that sufficient storage is available for the Unicenter TCPaccess address space.

### **521 "*pathname*" already exists**

Reason: A MKD command attempted to create a partitioned data set '*pathname*', but a cataloged data set already exists by that name.

### **530 Invalid userid or password, try again**

Reason: An invalid user ID or password has been entered.

### **530 Login required, enter USER**

Reason: The user is directed to supply a user ID to the host system.

### **530 Password expired, next time try: PASS *current-password/new-password***

Reason: The user's password expired. The user should update the password as shown.

### **530 Invalid new password, next time try: PASS *current-password/new-password***

Reason: The user's new password is invalid. Reenter a new password as shown.

### **530 FAILED ACCOUNTING EXIT**

Reason: Accounting exit ACEXIT00 rejected the signon attempt. ACEXIT00 is a local exit under control of the system administrator. See the *Customization Guide* for more information about ACEXIT00. The account was probably entered incorrectly.

Action: Retry the signon with a valid user ID/password/account combination. Contact your system administrator if the problem persists.

**530 Bad system security option**

Reason: A bad parameter list was passed to the external security system (ACF2, RACF, or Top Secret). The signon fails.

Action: Contact your Customer Support.

**530 No external security system is active**

Reason: A signon was attempted and the external security system (ACF2, RACF, or Top Secret) has become inactive.

Action: Contact your local external security system administrator.

**530 Password is not authorized for this userid**

Reason: A signon was attempted with an invalid password. The password was probably entered incorrectly.

Action: Retry the signon with a valid user ID/password combination.

**530 Password was omitted**

Reason: A signon was attempted without a password.

Action: Retry the signon with a valid user ID/password combination.

**530 User access has been revoked**

Reason: A signon attempt was rejected because your external security system (ACF2, RACF, or Top secret) has revoked your user ID.

Action: Contact your local external security system administrator.

**530 User access to the group has been revoked**

Reason: A signon attempt was rejected because your external security system (ACF2, RACF, or Top Secret) has revoked your user ID access to the group to which you are attempting to sign on.

Action: Contact your local external security system administrator.

### **530 User login rejected by installation exit routine**

Reason: A signon attempt was rejected by a local installation security exit.

Action: Contact your local external security system administrator.

### **530 User is not authorized to this application**

Reason: A signon attempt failed due to application security by your external security system (ACF2, RACF, or Top Secret).

Action: Contact your local external security system administrator.

### **530 User is not authorized to this terminal**

Reason: A signon attempt failed due to terminal security by your external security system (ACF2, RACF, or Top Secret).

Action: Contact your local external security system administrator.

### **530 Userid is not defined to the security system**

Reason: A signon was attempted using an unknown user ID. The user ID was probably entered incorrectly.

Action: Retry the signon with a valid user ID/password combination.

### **530 User is not defined to the group**

Reason: A signon was attempted using a group to which the user ID is not connected.

Action: Contact your local external security system administrator.

### **530 Userid matches userid of TCP address space**

Reason: A signon was attempted using the user ID of theUnicenter TCPaccess address space. As a security precaution, Unicenter TCPaccess does not accept its own user ID for a signon attempt.

### **550 Bad member name or generation index specified**

Reason: The name supplied for the member of a partitioned data set is incorrect.



**550 Catalog structure invalid or user lacks authority to catalog**

Reason: Dynamic allocation returned an information reason code of 5708. The attempted operation did not complete due to security reasons or an invalid catalog structure.

Action: Determine if the catalog is password protected or whether the catalog structure is invalid.

**550 Data set not found**

Reason: The file specified for processing cannot be located on the host system.

**550 Dataset medium is tape; request cannot be performed.**

Reason: A RENAME was requested but the data set is on magnetic tape.

**550 Error occurred during directory update, directory NOT updated**

Reason: A serious error occurs during ACDYNAL processing of the PDS directory.

**550 File access denied**

Reason: Access to the specified file is denied by the security (access control) system of the host.

**550 File cannot be accessed. HSM SVC is not supported by installation.**

Reason: File was migrated and HSM recall abnormally terminates with an x'16D'.

Action: Verify HSM is supported by the host operating system or HSM uses SVC 109 (x'6D'). Contact your local support personnel.

**550 File cannot be deleted. SCRATCH failed or expiration date not reached.**

Reason: A DELETE for a data set failed, either because the SCRATCH routine returned an error code, or because the expiration date has not occurred.

**550 File not accessed. A volume must be mounted, and mount is not permitted.**

Reason: A data set on a tape volume was requested, but tape processing was disallowed by the system administrator.

**550 File not accessed. Migrated file requires SITE RECALL command**

Reason: File has been migrated and HSM recall ability is not enabled for the user.

Action: Issue a SITE RECALL command to enable HSM for file transfer.

**550 New member name bad, format – DSN(OLDMEM) (NEWMEM)**

Reason: The new member name must be enclosed in parentheses () and must not be qualified (that is, no quotes).

**550 No multi-volume data sets**

Reason: FTP does not support multivolume data sets.

**550 Rename failed (New name already exists in directory)**

Reason: The new member name given in the RNTD command already exists as a member in the PDS.

**550 Rename failed (PDS DIRECTORY IS FULL)**

Reason: The STOW system command returned with an error indicating there were not enough directory blocks available to complete the rename.

**550 Rename failed (RNTD base name is invalid)**

Reason: A RENAME for a data set failed because the GDG base name is invalid.

**550 Rename failed (RNTD name CANNOT contain member name)**

Reason: A RENAME for a data set is issued but the new data set name specified a member name in addition to the data set name.

**550 Rename failed (RNFR data set not found)**

Reason: A RENAME for a data set was issued but the data set was not found.

**550 Rename failed (RNT0 data set exists in catalog already)**

Reason: A RENAME for a data set failed because a data set by that name already exists in the catalog.

**550 Rename failed (RNFR data set OBTAIN error)**

Reason: A RENAME for a data set failed due to an OBTAIN error.

**550 Rename failed (RNT0/RNFR CATLG error)**

Reason: A RENAME for a data set failed due to a catalog error.

**550 Rename failed (RNT0 name contained an invalid generation)**

Reason: A RENAME failed because the name contained an invalid generation.

**550 Not a partitioned dataset. Use DELE to delete**

Reason: A RMD command was received specifying a non-partitioned data set.

**550 Partitioned dataset contains members**

Reason: A RMD command was received for a partitioned data set that contains members. Delete the members first, or use DELE to delete the PDS.

**550 No matching datasets or members were found**

Reason: The file specified for processing cannot be located or no matching directory entries are found on the host system.

**550 MKD failed. DCBDSN data set *dcbdsn\_name* is invalid for a PDS.**

Reason: A SITE DCBDSN command was received, but the model data set has a record format (RECFM) or data set organization (DSORG) that is invalid for a PDS.

Action: Change the DCBDSN data set name, or issue SITE RESET to reset SITE parameters.

**Note:** The following replies will be prefixed with the number sequence 200- (rather than 500-), if the FTP configuration option SITEREPLY(200) is specified.

### 550 Unable to create unique data set name for STOU

Reason: A store unique (STOU) command was received. The server attempts to generate a unique name by appending 1-999 to the last qualifier of the data set name. It has exhausted the set of unique names.

Action: Change the file name

### 550 No matching entries were found *entries*

Reason: A directory command (LIST/NLIST) was received, but no entries were found that match the request. *entries* is one of the following:

members	If the current working directory is a partitioned data set.
volumes	If the filetype is VTOC.
data sets	In all other cases.

Action: Change the file name

### 550 Allocation of INTRDR failed.

Reason: Dynamic allocation of internal reader failed.

Action: Ensure you have enough INTRDRs available and Unicenter TCPaccess is not prevented from allocating them.

### 550 Submitted job can not be found.

Reason: Job submitted using PUTGET (automatic retrieval of a submitted job) could not be found in the JES queue.

Action: Check that the job submitted is not sent to another JES node.

### 550 No batch job resulted in submit.

Reason: GET using PUTGET (automatic retrieval of a submitted job) resulted in no batch jobs.

Action: Make sure the file you are submitting has a valid MVS job card.

**550 Timeout waiting for submitted job to end.**

Reason: Job submitted using PUTGET (automatic retrieval of a submitted job) did not complete within JESPUTGETTO limits.

**550 Job not found on JES queue**

Reason: The specified job cannot be found on the JES queue using the JESFILTER criteria.

**550 Unable to process now**

Reason: JES sysout file cannot be processed because it is currently being processed by a writer or another user.

**550 Authorization failed**

Reason: Access to the specified JES sysout file denied by the security (access control) system of the host.

**550 Requested JES file not found/unavailable**

Reason: The specified JES sysout file cannot be found or is currently being processed by a writer or another user.

**550 No held output for this job**

Reason: Job submitted using PUTGET (automatic retrieval of a submitted job) completed but contains no held sysout to retrieve.

**552 Insufficient space on volume. Transfer terminated**

Reason: The device to which the transferred file is being written has run out of storage space. The file transfer is incomplete.

**552 Unable to continue data transfer, data set full**

Reason: The receiving host does not have sufficient storage space for the data being transferred to it. The file transfer cannot continue.

### **553 Bad data set name syntax**

Reason: The file name specified is incorrect, or it violates the syntax or naming conventions defined by the remote host.

### **553 Indices must be between 1 and 8 characters**

Reason: The specified directory name cannot be longer than eight characters.

### **553 Operation failed – data set cataloged on another volume**

Reason: The file transfer completes unsuccessfully. A SITE command probably was issued to a specific VOLUME or UNIT and the file (data set) exists as a cataloged entry on another volume.

### **553 Operation failed – SITE command implies DISP=NEW – data set already exists**

Reason: One of the SITE commands (that is, SPACE, TRACK, CYLINDER, BLOCKS) implies that the data set should be created, but the data set already is defined.

### **554 Illegal RECFM in data set**

Reason: This message indicates one of these conditions:

- One of these record formats is specified as an unblocked print format: FA, FSA, VA, VSA, UA. The FTP Server can read from, but not write to, these files.
- The print format A is specified when the TYPE parameter specifies binary data. A print data set can be created or retrieved only as text, not as binary data.

### **554 LRECL or BLKSIZE invalid or inconsistent**

Reason: The logical record length and blocksize specified are either invalid or inconsistent.

### **554 Old data set not replaced, DSORG different**

Reason: Data transferred from the sending host does not replace the existing data set on the receiving host because the data set organizations conflict.

**554 SITE LRECL, BLKSIZE, or RECFM do not match those of existing data set**

Reason:                   The logical record length, blocksize, and/or record format specified in the preceding SITE commands are not consistent with the characteristics of the existing file on the receiving host.

**554 RETRIEve of a whole PDS is not permitted**

Reason:                   A RETR command was received specifying a partitioned data set as the pathname. A member name must be supplied.





# MAIL Messages

---

This chapter explains the causes of messages issued by the MAIL facilities in Unicenter TCPaccess and suggests appropriate responses.

It includes:

- [SNDMSG Messages](#) – Lists and describes all SNDMSG messages
- [SSMTP Messages](#) – SSMTP Receiver messages that start with the program ID SSMTP

## SNDMSG Messages

This section describes the SNDMSG messages.

### Access to Include File Denied file Unable to Find Data Set in Catalog

Reason: SNDMSG is unable to find the file specified in a .f= statement in the catalog. The parameter file represents the file name from the .f= statement.

Action: Verify that the file name is correct. Ensure that the file name is in the catalog.

### Access to Include File Denied . code reason:

Reason: The local security system denies read access to the file specified on a .f= statement. *code* is one of these reason codes from PACCESS: zero is successful, five is failed, eight is no memory, *reason* is a *reason* from the security system.

Action: Verify that the file name is correct. Ensure that you have read authority for the specified file.

### File Error reason in File *file*

- Reason: A file error was encountered while processing a .f= statement. *reason* is the reason returned from the file system; *file* is the file name
- Action: Repair or rebuild the file according to the reason given in the message.

### Recursion Error in File *FFFF*

- Reason: While processing a file (file) specified in a .f= statement, SNDMSG finds a line containing another .f= statement. This recursion is not supported.
- Action: Remove the .f= statement from the file. Manipulate the files separately and include them with multiple .f= statements to SNDMSG. Alternatively, edit and concatenate the files into a single file to be processed by a single .f= statement.

### Unable to Open File *file* Reason *reason*

- Reason: SNDMSG cannot process a .f= statement in a mail file because it cannot open the file. *file* is the file name from the .f= statement; *reason* is the reason returned from the file system
- Action: Verify that the file name exists and is in the catalog. Then use the reason information to determine the reason for the problem.

## SSMTP Messages

This section includes the SSMTP Receiver messages. All SSMTP Receiver messages start with program ID SSMTP.

### SSMTP MAIL FROM: *from* TO: *to*

- Reason: Message to inform the accounting log about a mail message from *from* to *to*. The message is written to the accounting log.

### SSMTP 214 -SAVE current session log

- Reason: Message indicates the current log is to be saved. The message is written to the error log.
- Action: This message normally follows other major error messages and indicates that the log data sets should be saved for later analysis.

**SSMTP 221 -QUIT: OK; operand ignored**

- Reason: SSMTP receives an RSET command with an operand that is ignored. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact software support for the host software.

**SSMTP 250 NOOP: operand ignored**

- Reason: SSMTP receives a NOOP command with an operand that is ignored. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact software support for the host software.

**SSMTP 354 Operand ignored; enter message**

- Reason: SSMTP receives a DATA command, but the message contains an operand that is ignored. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact software support for the host software.

**SSMTP 421 -Connection error: reason**

- Reason: SSMTP loses the connection with a foreign host. The reason (reason) for the failure is given. The message is written to the error log.
- Action: Check if the foreign host is down.

### **SSMTP 421 -Premature end-of file**

- Reason: SSMTP learns that it has lost its connection without receiving all data for the mail message. The message is written to the error log.
- Action: Check if the foreign host is down. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.

### **SSMTP 451 DATA aborted: CANNOT ALLOC MAIL FILE**

- Reason: SSMTP receives a DATA command, but SMTP is unable to allocate the mail file. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Dynamic allocation has failed. Check the volume that is in APPCFGxx's SMTP VOLUME= parameter. If there is no more room on the volume, reassemble APPCFGxx with a new volume specified on SMTP VOLUME= parameter. Retry the operation. If the error recurs or if there is plenty of space on the volume, contact Customer Support.

### **SSMTP 451 DATA aborted: CANNOT ALLOC TEMP FILE**

- Reason: SSMTP receives a DATA command, but SMTP is unable to allocate a temporary file for data. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Dynamic allocation has failed. Check the volume that is in the APPCFGxx SMTP VOLUME= parameter. If there is no more room on the volume, reassemble APPCFGxx with a new volume specified on SMTP VOLUME= parameter. Retry the operation. If the error recurs or if there is plenty of space on the volume, contact Customer Support.

### **SSMTP 451 I/O error; transaction aborted**

- Reason: SSMTP receives a DATA command with an I/O error. The mail message is discarded. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software or hardware is misbehaving. Contact Customer Support or hardware support for the failing component.

**SSMTP 451 Mailer error; transaction aborted**

- Reason: SSMTP receives a DATA command, but an error is received during the data transfer and the operation is aborted. Transaction is aborted, no message is recorded. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.

**SSMTP 451 VRFY Cannot load user table**

- Reason: SSMTP receives a VRFY command, but the SMTPUSR table is not loaded. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. C The SMTPUSR csect cannot be located. Check the STEPLIB load libraries for module SMTPUSR. The Customization Guide describes how to create the module. Retry the operation. Bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

**SSMTP 452 MAIL rejected: ran out of storage**

- Reason: SSMTP receives a MAIL command, but a shortage of storage is detected. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. If there is a proliferation of the same kind of storage, a PTASK may be allocating storage in a loop. Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. Raise the region size on the Unicenter TCPaccess job. Retry the operation again. If the error recurs, contact Customer Support.

**SSMTP 500 Blank/empty input line: ignored**

- Reason: SSMTP receives a blank or empty reply line. The command is ignored. The message is written to the error log.

### **SSMTP 500 command rejected: command line too long**

- Reason: SSMTP receives a bad reply; the rejected command (command) is given. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.

### **SSMTP 500 command rejected: no such command**

- Reason: SSMTP receives an invalid command (command). The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.

### **SSMTP 501 HELO rejected: domain too long**

- Reason: SSMTP receives a HELO command, but Domain name is larger than 65 bytes. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the HELO command sent across the connection. Contact Customer Support for the host software.

**SSMTP 501 HELO rejected: invalid domain**

- Reason: SSMTP receives a HELO command, but the Domain name is invalid. The command is ignored. The message is written to the error log.
- Action: Verify Domain name from remote host. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the HELO command sent across the connection. Contact software support for the host software.

**SSMTP 501 HELO rejected: no domain given**

- Reason: SSMTP receives a HELO command, but no Domain name is found in the X-FROM field. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a HELO command without any arguments. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the HELO command sent across the connection. Contact Customer Support for the host software.

**SSMTP 501 MAIL rejected: FROM: missing**

- Reason: SSMTP receives a MAIL command, but the FROM keyword is not found. The command is ignored. The message is written to the error log.
- Action: The remote host may be transmitting an invalid command. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the MAIL command sent across the connection. Contact Customer Support for the host software.

### **SSMTP 501 MAIL rejected: invalid path**

- Reason: SSMTP receives a MAIL command with an invalid path. The command is ignored. The message is written to the error log.
- Action: The remote host may be transmitting an invalid path. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the MAIL command sent across the connection. Contact Customer Support for the host software.

### **SSMTP 501 MAIL rejected: no path given**

- Reason: SSMTP receives a MAIL command, but no path or operands are found. The command is ignored. The message is written to the error log.
- Action: The remote host may be transmitting an invalid command. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start an TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the MAIL command sent across the connection. Contact Customer Support for the host software.

### **SSMTP 501 MAIL rejected: path too long**

- Reason: SSMTP receives a MAIL command, but the path is larger than 256. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the MAIL command sent across the connection. Contact Customer Support for the host software.



**SSMTP 501 RCPT rejected: invalid path**

- Reason: SSMTP receives an RCPT command, but the path is invalid. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

**SSMTP 501 RCPT rejected: no path given**

- Reason: SSMTP receives an RCPT command, but no path or operand is found. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a RCPT command without any path specified. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

**SSMTP 501 RCPT rejected: path can't be '<>'**

- Reason: SSMTP receives an RCPT command with an invalid path name of '<>'. The command is ignored. The message is written to the error log.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

### **SSMTP 501 RCPT rejected: path name too long**

**Reason:** SSMTP receives an RCPT command, but the path name is greater than 256. The command is ignored. The message is written to the error log.

**Action:** Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

### **SSMTP 501 RCPT rejected: TO: missing**

**Reason:** SSMTP receives an RCPT command, but no TO: field is found. The command is ignored. The message is written to the error log.

**Action:** Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

### **SSMTP 501 RCPT rejected: user too long or missing**

**Reason:** SSMTP receives an RCPT command with an invalid mailbox ID. The command is ignored. The message is written to the error log.

**Action:** The remote host transmitted an invalid user ID. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. Examine the RCPT TO command sent across the connection. If the error recurs, try to determine from the trace which host software is misbehaving. Check that the RCPT TO command is in the correct format. Contact Customer Support for the host software.

**SSMTP 501 RFY rejected: no user given**

- Reason: SSMTP receives a VRFY command, but no User operand is received. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a command without any arguments. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Examine the VRFY command sent across the connection. Contact Customer Support for the host software.

**SSMTP 502 command rejected: command not implemented**

- Reason: SSMTP receives an unimplemented command (SEND, SOML, SAML, EXPN, or TURN). The command is ignored. The message is written to the error log.
- Action: Save output from the Unicenter TCPaccess job. The remote host may be transmitting an unsupported SMTP command. Contact Customer Support.

**SSMTP 503 DATA rejected: must give recipient(s) first**

- Reason: SSMTP receives a DATA command, but no RCPT command preceded the DATA command. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a command out of sequence or the previous command was invalid. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Determine from the trace if the RCPT command was sent. Contact Customer Support for the host software.

### **SSMTP 503 DATA rejected: no transaction in progress**

- Reason: SSMTP receives a DATA command, but no transaction is in progress. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a command out of sequence or the previous command was invalid. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.

### **SSMTP 503 RCPT rejected: no transaction in progress**

- Action: SSMTP receives an RCPT command, but no MAIL command preceded the RCPT. The command is ignored. The message is written to the error log.
- Action: The remote host transmitted a command out of sequence. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Determine if a MAIL command was sent before the RCPT command. Contact Customer Support for the host software.

### **SSMTP 550 User name unknown/invalid**

- Reason: SSMTP receives a VRFY command, but the user is invalid or not in correct form for a TSO user ID. The command is ignored. The message is written to the error log.
- Action: Save output from the Unicenter TCPaccess job. The remote host transmitted an invalid user ID. If the user does not exist, ignore this message. If this is a valid user, check your SMTPUSR exit code. Retry the operation. If the problem persists, contact Customer Support.

**SSMTP 554 Message was empty; transaction aborted**

**Reason:** SSMTP receives a DATA command, but no data is received. No message is recorded. The message is written to the error log.

**Action:** The local host received no data for the current mail transaction. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support for the host software.



# Error Messages

---

This chapter contains the perror messages issued by Unicenter TCPaccess. It includes sections describing perror messages and socket library error messages.

The perror messages included in the perror module in POPEN are described in the following sections:

- [Network Error Messages](#)
- [Disk perror Messages](#)
- [Other Error Messages](#)
- [Mail Error Messages](#)
- [User Configuration or Subsystem Error Messages](#)
- [Socket Library perror Messages](#)

The perror function generates a message of this form:

user supplied string if any: socket library generated string

The perror message connect includes a user-supplied string, and operation already in progress is the socket library generated string:

connect : Operation already in progress

## Network Error Messages

This section lists all network error messages.

### Bad Source Route

Reason:	The option field in the IP datagram requests source routing. A list of a sequence of machines that the datagram must visit is in error.
Action:	If the host or an intermediate network is down, contact your proper support personnel to bring the host or network back up. If the problem is an unknown or undefined host, check that the host is in the Unicenter TCPaccess Host Names Table. Insert a host or network entry if necessary and reassemble the Host Names Table. Ignore this message if the user inadvertently entered a nonexistent host.

### Host Unreachable

Reason:	The host is down or does not exist.
Action:	If the host or an intermediate network is down, contact your proper support personnel to bring the host or network back up. If the problem is an unknown or undefined host, check that the host is in the Unicenter TCPaccess Host Names Table. Insert a host or network entry if necessary and reassemble the Host Names Table. Ignore this message if the user inadvertently entered a nonexistent host.

### Interrupt Received

Reason:	An interrupt is received.
Action:	None.

### Local Net Down

Reason:	The local network is down.
Action:	None.



### Net Unreachable

- Reason: The network requested is not reachable.
- Action: If the host or an intermediate network is down, contact your proper support personnel to bring the host or network back up. If the problem is an unknown or undefined host, check that the host is in the Unicenter TCPaccess Host Names Table. Insert a host or network entry if necessary and reassemble the Host Names Table. Ignore this message if the user inadvertently entered a nonexistent host.

### Network Timeout

- Reason: A network time out occurs.
- Action: Check that the network node addressed is on the network and in the Host Names Table (ARPINAMS) and APPCFGxx. Ignore this message if the network addressed does not exist on your network. Monitor these occurrences. If they continue, notify your network specialist for further analysis and identification of the offending remote host.

### No Buffer Storage

- Reason: The event fails due to a lack of storage.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. If there is a lot of the same kind of storage, a ptask may be looping storage. Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. Raise the region size on the Unicenter TCPaccess job. Retry the operation again. If the error recurs, contact Customer Support.

### Port Unreachable

- Reason: The requested port is unavailable at this time.
- Action: Check the WTO and JOB logs for related messages to discover the PORT and the host that are unavailable. Verify that the host supports the protocol for which the PORT is used. List the user commands that generated this error. Start up a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### Retransmitting

- Reason: A TCP acknowledgment for a segment is not received and TCP retransmits the segment.
- Action: Monitor these occurrences. If they continue, notify your network specialist for further analysis and identification of the offending remote host.

### Software Error!

- Reason: One of the programs performing this service ABENDED.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action. Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

### Source Quench

- Reason: An ICMP source quench is sent requesting that the service stop or reduce the transmission of IP datagrams.

### TCP Conn Closed

- Reason: The TCP connection is closed.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### TCP Conn Reset

- Reason: A condition occurs that forces the software to break the connection.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error. Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

**Unknown TCP Error**

- Reason: An unknown TCP error occurs.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error.
- Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

## Disk perror Messages

This section contains all disk perror messages.

**Access control violation**

- Reason: A user requests a resource to which the user does not have access. Access is denied.

**DAIR error**

- Reason: The task fails in dynamic allocation of a resource. This is an SVC99 error.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action. Use the DAIR code to direct your actions.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails, bring it down and back up again. If the error recurs, contact Customer Support.

**Dataset not found**

- Reason: A request is made to read a nonexistent data set.
- Action: Retry the task asking for the correct data set. Check to make sure the catalog actually points to a data set. If there is no data set, fix the catalog.

### **DSN syntax error**

Reason: A request was made for a data set name that has illegal syntax.

Action: Retry the task using a valid data set name.

### **ENQ conflict**

Reason: A request is made for a data set owned by another user.

Action: Retry the task later.

### **Member not found**

Reason: A request is made to read a nonexistent PDS member.

Action: Retry the task asking for the correct member.

### **Not enough disk space**

Reason: Either the disk does not have enough space, or the data set has reached its maximum number of extents.

Action: Retry the task asking for more space or an alternate volume that has more space.

### **Not enough real storage**

Reason: Unicenter TCPaccess runs out of real storage.

Action: Save all output from the V job. Check the WTO and JOB logs for related messages. If there is a lot of the same kind of storage, a ptask may be allocating storage in a loop.

Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. Raise the region size on the Unicenter TCPaccess job. Retry the operation again. If the error recurs, contact Customer Support.

**Not enough TIOT space**

- Reason: The task runs out of core for the TIOT.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. If there is a lot of the same kind of storage, a ptask may be allocating storage in a loop.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails, bring it down and back up again. Raise the region size on the Unicenter TCPaccess job. Retry the operation again. If the error recurs, contact Customer Support.

**PDYNAL/IDYNAL bug**

- Reason: The Unicenter TCPaccess dynamic allocation routines ABEND. The task cannot be completed.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails, bring it down and back up again. If the error recurs, contact Customer Support.

**POPEN failed**

- Reason: The Unicenter TCPaccess POPEN utility to open a dcb (or ACB) fails.
- Action: Check for a correct call to POPEN. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

**Software error**

- Reason: The programs performing the service ABEND. The task cannot be completed.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

### **SVC99 parm error**

- Reason: The task sends bad parameters to Unicenter TCPaccess dynamic allocation routines.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

### **Unknown disk error**

- Reason: An unknown disk error occurs.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages that may direct the correct course of action.
- Retry the operation first on the current version of Unicenter TCPaccess. If it fails again, bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

### **VOL/Unit unavailable**

- Reason: The user requests access to a disk volume or unit that is unavailable.
- Action: Check that the volume or unit is online. If the user requested a nonexistent volume or unit, ignore this message.

## Other Error Messages

This section lists other miscellaneous error messages.

### **I/O error cc**

Reason: I/O error occurs with a condition code of cc.

Action: Save all output from the job. Contact Customer Support.

### **No free storage!**

Reason: Unicenter TCPaccess runs out of pcore storage.

Action: Increase the region size for Unicenter TCPaccess. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages.

Retry the operation first on the current version of Unicenter TCPaccess. Retry the operation again. If it fails again, bring Unicenter TCPaccess down and back up again. Raise the region size on the Unicenter TCPaccess job. If the error recurs, contact Customer Support.

## Mail Error Messages

This section contains common error messages from mail.

### **Died during transmission**

Reason: The Remote host goes down during transmission.

Action: Check that the remote host is running and able to receive mail. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error.

Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### Embedded file recursion

Reason: USMTMP discovers a recursive unsupported service within Unicenter TCPaccess.

Action: Check related messages in the logs for proper response. Save all output from the Unicenter TCPaccess job. Contact Customer Support.

### Embedded file security failed

Reason: The Mailer encounters a .f= statement specifying an invalid data set name, a data set name that is not cataloged, or the name of a file that the local security system (for example., RACF, ACF2, CA-TOPSECRET) will not let the Mailer read.

Action: Correct the data set name; verify that it is in the catalog accessible to Unicenter TCPaccess and that the local security system provides read authority to Unicenter TCPaccess for that file.

This mail message must have been entered from SPOOL#3 or SPOOL#4.  
SNDMSG now expands .f= statements itself.

### Embedded file unknown

Reason: USMTMP cannot get to a file to perform the mail service.

Action: Check related messages in the WTO logs for proper response.

### File open fail!!

Reason: Cannot open the mail file data set.

Action: Check related messages in the logs for the data set name. Examine the data set. Contact Customer Support.

### Header parsing error

Reason: USMTMP has trouble parsing either the X-FROM or X-TO address field.

Action: Check related messages in the WTO logs for the data set name. Examine the data set. Check the X-FROM and X-TO headers for errors. Contact Customer Support.



**Host down**

- Reason: The remote host is down.
- Action: Check that the remote host is running and able to receive mail.

**Host look up error**

- Reason: Unicenter TCPaccess encounters a problem looking up the host.
- Action: Check the Unicenter TCPaccess WTO logs for the mail data set name. Check that the host in the message is on the network. Save all output from the Unicenter TCPaccess job. Get the list of commands entered by the user that generated this error.
- If the problem is an unknown or undefined host, check that the host is in the Unicenter TCPaccess Host Names Table. Insert the needed host entry if necessary and reassemble the Host Names Table. Ignore this message if the user inadvertently entered a nonexistent host.
- Retry the operation. If the problem persists, contact Customer Support.

**Internal format error**

- Reason: An error is encountered with either the X-FROM or X-TO header.
- Action: Check related messages in the WTO logs for the data set name. Examine the data set. Check the X-FROM and X-TO headers for errors. Contact Customer Support.

**Internal format error Bad fmuid length**

- Reason: The user ID in the X-TO field is longer than eight characters and cannot be a user ID on an IBM system. This error message should appear only if the MUNGE(YES) parameter is specified on the SMTP statement in APPCFGxx.
- Action: Check related messages in the logs for the data set name. Examine the data set name and userid. Examine the X-TO header for errors. Contact Customer Support.

**Internal format error Bad XFROM**

- Reason: An error is encountered with the X-FROM header.
- Action: Check related messages in the logs for the data set name. Examine the data set. Examine the X-FROM header for errors. Contact Customer Support.

### **Internal format error Bad XTO**

Reason: An error is encountered with the X-TO header.

Action: Check related messages in the logs for the data set name. Examine the data set and the X-TO header for errors. Contact Customer Support.

### **Internal format error No XFROM**

Reason: No X-FROM header exists.

Action: Check related messages in the logs for the data set name. Examine the data set for an X-FROM header. Contact Customer Support.

### **Internal format error No XTO**

Reason: No X-TO header exists.

Action: Check related messages in the logs for the data set name. Examine the data set for an X-TO header. Contact Customer Support.

### **I/O Error reading disk file!!**

Reason: An I/O error occurs when USMTP tries to read the mail data set on disk.

Action: Check related messages in the logs for the data set name. Examine the data set. Use ICKDSF and/or EREP to find any problems on the tracks where the mail data set is stored. Contact Customer Support.

### **No such mailbox**

Reason: The Remote sends a 55x reply. This can indicate either bad mailbox name or that the retry errors maximum was exceeded.

Action: Check to make sure that the mailbox name is on the remote host.

**No user table!!**

- Reason: This message may be generated due to one of the following conditions:
- Module SMTPUSR contains no users or does not exist.
    - Save all output from the Unicenter TCPaccess job.
    - Check the WTO and JOB logs for related messages.
  - The SMTPUSR csect cannot be located.
    - Check the ACFETCH load libraries for module SMTPUSR.
    - See the Unicenter TCPaccess *Customization Guide* for information on how to create the SMTPUSR module.
- Action: Retry the operation. Bring Unicenter TCPaccess down and back up again. If the error recurs, contact Customer Support.

**Not local user**

- Reason: The remote host states that the user is not a valid user on its system.
- Action: Check the validity of the host and user name for the remote host.

**Overflow at remote site**

- Reason: The data transfer overflows the remote hosts buffers.
- Action: Retry the transfer.

**SMTP protocol error**

- Reason: The Remote host believes a protocol error occurred.
- Action: Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error.
- Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### **Syntax error in mailbox name**

- Reason: The Remote host encounters an error in the mailbox name.
- Action: Check other messages in the logs for related messages. Make sure that the mail message specifies a valid mailbox name.

### **TCP connection refused**

- Reason: The remote host refuses to accept a session with Unicenter TCPaccess.
- Action: Check the error logs for related messages. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error.
- Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### **Truncated spool file**

- Reason: The mail file data set is incomplete. A premature end-of-file condition occurred.
- Action: Check related messages in the logs for the data set name. Examine the data set. Contact your proper support personnel or Customer Support.

### **Unexpected reply from remote SMTP**

- Reason: The Remote host sends an unexpected reply to Unicenter TCPaccess.
- Action: Check related messages in the logs for proper response. Save all output from the Unicenter TCPaccess job. Check the WTO and JOB logs for related messages. Get the list of commands entered by the user that generated this error.
- Start a TCPEEP trace using the TELNET, DEBUG, and SYSOUT options. Retry the operation. If the error recurs, try to determine from the trace which host software is misbehaving. Contact Customer Support.

### Unknown host

Reason: The Remote host is unknown.

Action: Save all output from the Unicenter TCPaccess job. Get the list of commands entered by the user that generated this error.

If the problem is an unknown or undefined host:

- Check that the host is in the Unicenter TCPaccess Host Names Table.
- Insert the needed host entry if necessary and reassemble the Host Names Table.
- Ignore this message if the user inadvertently entered a nonexistent host.
- Retry the operation.

If the problem persists, contact Customer Support.

## User Configuration or Subsystem Error Messages

This section lists all the messages that might occur as subclasses to the main message, User Configuration or Subsystem Error.

Messages are listed starting with the most general message.

### User Configuration or Subsystem Error

Reason: This error message is generated when a user issues a socket() function call and either the socket configuration is invalid or the socket library encounters a problem when trying to establish a session on behalf of the user with the API. This error message may be followed by another line that provides more insight into the exact cause of the problem. This extra line is generated if the sockcfg.flags element of the socket configuration structure has the EXTERRNOMSG bit set.

errno: ECONFIG

Action: Set the EXTERRNOMSG bit on in the socket configuration and rerun the program. A more descriptive error message should then be generated and the action listed under that message taken.

**Access method CVT unavailable User Configuration or Subsystem Error Access method CVT unavailable**

Reason: The access method CVT is unavailable at AOPEN time.

errno: ECONFIG / EAPCBECVT

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**AMODE inconsistent with AOPEN User Configuration or Subsystem Error AMODE inconsistent with AOPEN**

Reason: The address mode has changed between the time the AOPEN was issued and the time the ACLOSE was issued by the socket library to the API.

errno: ECONFIG / EAPCBEAMD

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**AMTV validity check error User Configuration or Subsystem Error AMTV validity check error**

Reason: The access method transfer vector causes an error at either AOPEN or ACLOSE time.

errno: ECONFIG / EAPCBETRV

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**AOPEN/ACLOSE not issued from PRB User Configuration or Subsystem Error AOPEN/ACLOSE not issued from PRB**

Reason: The AOPEN issued by the socket library is done when the socket library is not running under a PRB.

errno: ECONFIG / EAPCBEPB

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**APCB already closed User Configuration or Subsystem Error APCB already closed**

Reason: The APCB issued on the ACLOSE by the socket library to the API is already marked as closed by the API.

errno: ECONFIG / EAPCBECLS

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**APCB already opened User Configuration or Subsystem Error APCB already opened**

Reason: The APCB used by the socket library to open a user session with the API is listed as open by the API.

errno: ECONFIG / EAPCBEOPN

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **APCB has permanent error User Configuration or Subsystem Error APCB has permanent error**

Reason: The APCB used by the socket library to initiate and terminate a user session with the API has been marked with a permanent error by the API.

errno: ECONFIG / EAPCBEPER

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **APCB is busy User Configuration or Subsystem Error APCB is busy**

Reason: The APCB used by the socket library to initiate and terminate user sessions with the API is detected busy by the API at either AOPEN or ACLOSE time.

errno: ECONFIG / EAPCBESY

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **APCB validity check error User Configuration or Subsystem Error APCB validity check error**

Reason: The APCB used by the socket library to perform an AOPEN to the API is invalid.

errno: ECONFIG / EAPCBEVCK

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **API application close failed User Configuration or Subsystem Error API application close failed**

Reason: During socket library termination, the ACLOSE issued to the API fails for an unspecified error.

errno: ECONFIG / EACLOSEFAILED

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.



**API application open failed User Configuration or Subsystem Error API application open failed**

- Reason: When opening a session with the API, the socket library encounters an error about which it cannot be more specific.
- errno: ECONFIG / EAOPENFAILED
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**Application ID too long User Configuration or Subsystem Error Application ID too long**

- Reason: The user's socket configuration specifies an application ID that contains more than eight characters.
- errno: ECONFIG / EAPPLIDTOOLONG
- Action: Correct the application ID parameters of the socket configuration, recompile it, and relink the application.

**Application ID too short User Configuration or Subsystem Error Application ID too short**

- Reason: The user's socket configuration specifies an application ID to be used on the opening of the user's session with the API subsystem that had zero characters.
- errno: ECONFIG / EAPPLIDTOOSHORT
- Action: Correct the application ID parameters of the socket configuration, recompile it, and relink the application.

**Atexit error User Configuration or Subsystem Error Atexit error**

- Reason: The user's socket configuration has an error with the library termination exit function.
- errno: ECONFIG / EEXITERROR
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application.

### **Cannot establish API session User Configuration or Subsystem Error Cannot establish API session**

- Reason: The socket library cannot establish a user session with the API for an unknown reason.
- errno: ECONFIG / EAPCBELEG
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **Cannot initialize user environment User Configuration or Subsystem Error Cannot initialize user environment**

- Reason: The API cannot initialize the user environment at AOPEN time.
- errno: ECONFIG / EAPCBEENV
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **Cannot release API session User Configuration or Subsystem Error Cannot release API session**

- Reason: At socket library termination time, the user session cannot be closed due to an internal error with the API.
- errno: ECONFIG / EAPCBEEND
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **Close timeout value too big User Configuration or Subsystem Error Close timeout value too big**

- Reason: The close time-out value of the socket library's configuration is invalid.
- errno: ECONFIG / ECLOSETIMEOUT
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application.

**Duplicate session for user User Configuration or Subsystem Error Duplicate session for user**

Reason: A duplicate user session with the API is detected when the socket library issues an AOPEN to the API.

errno: ECONFIG / EAPCBEDUP

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**Environment memory free error User Configuration or Subsystem Error Environment memory free error**

Reason: During socket library termination the socket library receives an error when trying to free storage that it had allocated.

errno: ECONFIG / EENVRFREE

Action: Determine the proper operation of the C runtime library.

**Errno base error User Configuration or Subsystem Error Errno base error**

Reason: The user's socket configuration has an error with the errno base value.

errno: ECONFIG / EERRNOBASE

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application.

**Internal logic error User Configuration or Subsystem Error Internal logic error**

Reason: The API encounters an internal logic error while attempting to complete the AOPEN issued by the socket library to initiate a user session with the API.

errno: ECONFIG / EAPCBELER

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **Invalid access method version User Configuration or Subsystem Error Invalid access method version**

- Reason: The version number of the APCB is not the same as the API.
- errno: ECONFIG / EAPCBEVER
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

### **Invalid option on AOPEN/ACLOSE User Configuration or Subsystem Error Invalid option on AOPEN/ACLOSE**

- Reason: The option field of the APCB is invalid when the socket library issues the AOPEN to the API.
- errno: ECONFIG / EAPCBEOPT
- Action: Review the installation procedure of the socket library. Particular attention should be paid to the socket configuration variable that the socket library stuffs into the APCB on the AOPEN. Correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, verify the proper operation of the API and Unicenter TCPaccess.

### **No memory for AOPEN User Configuration or Subsystem Error No memory for AOPEN**

- Reason: The API lacks the proper amount of storage to initiate another user session.
- errno: ECONFIG / EAPCBEMEM
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, the user should verify the proper operation of the API and Unicenter TCPaccess.

**No memory for environment initialization User Configuration or Subsystem Error No memory for environment initialization**

Reason: During socket initialization the socket library cannot allocate storage for its environment initialization routine.

errno: ECONFIG / EENVRALLOC

Action: Try to find out why there is a shortage of storage.

**No memory for user initialization User Configuration or Subsystem Error No memory for user initialization**

Reason: During socket library initialization, the socket library cannot allocate enough storage to initialize properly.

errno: ECONFIG / EUSERALLOC

Action: Try to find out why there is a shortage of storage.

**No subsystem ID configured User Configuration or Subsystem Error No subsystem ID configured**

Reason: The user's socket configuration does not specify a subsystem ID to use when opening a session with the assembler API subsystem.

errno: ECONFIG / ENOSUBSYSID

Action: Correct the subsystem ID parameter of the socket configuration, recompile it, and relink the application.

**Password too long User Configuration or Subsystem Error Password too long**

Reason: The user's socket configuration specifies an application password that contains more than eight characters.

errno: ECONFIG / EPASSWDTOOLONG

Action: Correct the application password parameters of the socket configuration, recompile it, and relink the application.

### **Password too short User Configuration or Subsystem Error Password too short**

Reason: The user's socket configuration specifies an application password that contains zero characters.

errno: ECONFIG / EPASSWDTOOSHORT

Action: Correct the application password parameter of the socket configuration, recompile it, and relink the application.

### **Service ID too long User Configuration or Subsystem Error Service ID too long**

Reason: The user's socket configuration specifies a service ID that is greater than eight characters.

errno: ECONFIG / ESVCIDTOOLONG

Action: Correct the service ID parameters of the socket configuration, recompile it, and relink the application.

### **Service ID too short User Configuration or Subsystem Error Service ID too short**

Reason: The user's socket configuration specifies a service ID of zero bytes.

errno: ECONFIG / ESVCIDTOOSHORT

Action: Correct the service ID parameters of the socket configuration, recompile it, and relink the application.

### **Signal initialization failed User Configuration or Subsystem Error Signal initialization failed**

Reason: The signal parameters of the socket configuration are invalid.

errno: ECONFIG / ESIGNAL

Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application.

**Socket buffering error User Configuration or Subsystem Error Socket buffering error**

- Reason: The user's socket configuration has an invalid setting in the buffering configuration parameters.
- errno: ECONFIG / EBUFFERVALUES
- Action: Review the installation procedure of the socket library, correct any configuration information with the socket configuration, recompile it, and relink the application.

**Socket descriptor error User Configuration or Subsystem Error socket descriptor error**

- Reason: The user's socket configuration has an invalid setting dealing with the assignment of socket descriptors.
- errno: ECONFIG / ESOCKETVALUES
- Action: Review the installation procedure of the socket library, make any necessary corrections to the socket configuration, recompile it, and relink the application.

**Subsystem halted User Configuration or Subsystem Error Subsystem halted**

- Reason: The API subsystem is halted for some reason.
- errno: ECONFIG / EAPCBEHLT
- Action: Investigate the reason the API is halted and retry the program once it has been restarted.

**Subsystem ID too long User Configuration or Subsystem Error Subsystem ID too long**

- Reason: The user's socket configuration specifies a subsystem ID of more than four bytes to use when opening a session with the assembler API subsystem.
- errno: ECONFIG / ESUBSYSIDTOOLONG
- Action: Correct the subsystem ID parameter of the socket configuration, recompile it, and relink the application.

### **Subsystem ID too short User Configuration or Subsystem Error Subsystem ID too short**

Reason: The user's socket configuration specifies a subsystem ID of zero bytes to use when opening a session with the assembler API subsystem.

errno: ECONFIG / ESUBSYSIDTOOSHORT

Action: Correct the subsystem ID parameter of the socket configuration, recompile it, and relink the application.

### **Subsystem is terminating User Configuration or Subsystem Error Subsystem is terminating**

Reason: The API subsystem is terminating execution. No more users can open sessions with it.

errno: ECONFIG / EAPCBTERM

Action: Determine why the API is terminating and then retry the program once it is running again.

### **Subsystem not active User Configuration or Subsystem Error Subsystem not active**

Reason: The subsystem of the API is not active.

errno: ECONFIG / EAPCBEACT

Action: Have the operations personnel start the API and Unicenter TCPaccess.

### **Subsystem not configured User Configuration or Subsystem Error Subsystem not configured**

Reason: The subsystem of the API is not configured.

errno: ECONFIG / EAPCBCFG

Action: Review the installation procedure of the socket library. Pay particular attention to the subsystem ID. Correct any configuration information with the socket configuration, recompile it, and relink the application. If no configuration errors are found, verify the proper operation of the API and Unicenter TCPaccess.



**Subsystem not initialized User Configuration or Subsystem Error  
Subsystem not initialized**

Reason: The subsystem of the API and Unicenter TCPaccess is not initialized at the current time.

errno: ECONFIG / EAPCBERDY

Action: Verify that the API is running and has properly initialized before retrying the program.

**User memory free error User Configuration or Subsystem Error  
User memory free error**

Reason: At socket library termination, the socket library encounters an error when trying to free storage that it had allocated.

errno: ECONFIG / EUSERFREE

Action: Verify that the C runtime library is functioning properly.

## Socket Library perror Messages

This section lists all Socket Library perror messages in alphabetical order.

**Address already in use**

Reason: The user requests that a particular port number be used locally but this port is being used by another.

errno: ESADDRINUSE

Action: Select another port number or try again later.

**Address family not supported by protocol family**

Reason: The socket user issues a call that requires a socket name but specifies a socket domain other than AF\_INET.

errno: ESAFNOSUPPORT

Action: Ensure that the domain of a socket name specifies AF\_INET.

### Bad address

Reason: The user passed a pointer to the socket library that points to an area of storage that cannot be accessed by the socket library.

errno: EFAULT

Action: Verify that the pointers used on the socket function call are valid.

### Bad socket number

Reason: A socket call is issued by the user specifying a socket descriptor that is not active. All socket functions have the socket descriptor as their first argument. The socket descriptor is the small integer returned by the socket function.

errno: ESBADF

Action: Ensure that a proper socket descriptor is passed when this call is issued.

### Broken pipe

Reason: A socket that had its transmit simplex shutdown had a write request of some form issued to it. Write requests include `send()`, `sendto()`, `sendmsg()`, `write()`, and `writev()`.

errno: ESKPIPE

Action: Change this programming tactic.

### Can't assign requested address

Reason: The name passed by a user to the socket library is invalid. Typically either the port is zero or the address equals `INADDR_ANY`.

errno: ESADDRNOTAVAIL

Action: Verify that a valid socket name is being passed to the failing function.

### Can't send after socket shutdown

Reason: After shutting down a socket, the user attempts to transfer data. This error message is generated because the error code is never set by the socket library.

errno: ESHUTDOWN

### Connection refused

Reason: The connect request initiated by a connect() function call is refused by the remote system.

errno: ECONNREFUSED

Action: Verify that the connect request is formatted properly and if the failure still occurs, verify that the remote system is functioning properly.

### Connection reset by peer

Reason: The remote endpoint reset the connection for some unknown reason.

errno: ECONNRESET

Action: Check the remote system and then try to solve the problem.

### Connection timed out

Reason: The connection has timed out. The socket should be closed and then recreated and the connection reestablished.

errno: ETIMEDOUT

Action: Close the socket and then reopen it and re-establish the connection.

### Destination address required

Reason: A send request of some form is issued on a socket operating in connectionless mode but the user does not pass the socket library the name of the remote endpoint to send the data to.

errno: EDESTADDRREQ

Action: Either use a connection or association-base socket or ensure that either the function sendto() or sendmsg() be used and a name of the remote endpoint specified.

### Destination unreachable

- Reason: The remote destination is currently unreachable.
- errno: EDESTUNREACH
- Action: Determine the network problem that made the destination unreachable; then try again.

### Host is down

- Reason: The remote host to which the user is trying to establish a connection or transfer data to and from is not currently operating on the network. This error message is generated because the error code is never set.
- errno: EHOSTDOWN
- Action: Wait for the system to become operational.

### Host is unreachable

- Reason: The remote host of a connection or the remote host to which data is being transferred is unreachable.
- errno: EHOSTUNREACH
- Action: Ensure this is the case and then wait for the remote system to become reachable via the network.

### I/O error

- Reason: A socket call encounters an I/O error. This error message should never be generated because the error code is never set.
- errno: ESIO

### Interrupted system call

Reason: A system call is interrupted. This error message should never be issued because the error code is never set.

errno: ESINTR

### Invalid argument

Reason: This error message is generated for a number of reasons. The reasons are listed in the *Assembler API Macro Reference* in the section covering each function.

errno: ESINVAL

Action: Find out which socket function is being called at the time of the error and refer to the *Assembler API Macro Reference* for specific information.

### Message too long

Reason: The user issues a send request of some form on a socket that maintains message boundaries but the length of the message exceeds either the maximum allowable message or the maximum allowable message based on the current buffer space allocations.

errno: EMSGSIZE

Action: Either decrease the size of the send request or, if buffer allocations are the restricting factor, increase buffer allocation.

### Network dropped connection on reset

Reason: A network problem forces the connection to the remote endpoint to be terminated. This error message is never generated because the error code is not set by the socket library.

errno: ESNETRESET

Action: Determine the problem with the network and retry.

### Network flow control

Reason: The network is congested and the local user should slow his transmission rate to help alleviate the congestion.

errno: ESNETOVRUN

Action: Adjust the data transmission rate to lessen network congestion.

### Network is down

Reason: The local network interface is not operational. This error message is never generated because the error code is not set by the socket library.

errno: ESNETDOWN

Action: The local network interface must be placed online and a retry initiated.

### Network is unreachable

Reason: The remote endpoint cannot be reached because the remote network is unreachable. This error message is never generated because the error code is not set by the socket library.

errno: ESNETUNREACH

Action: Check that correct remote endpoint is being used. If so, try to find out why the remote network is unreachable.

### No buffer space available

Reason: The function called cannot acquire the proper amount of buffer space necessary to complete execution of the request. The request is aborted abnormally.

errno: ESNOBUFFS

Action: Try to determine the reason for a lack of storage. This type of problem may be caused by a piece of software that is allocating storage and never freeing it.

**No error**

Reason: No error occurs within the socket library.

**Not enough memory**

Reason: Storage cannot be allocated to complete the request.

errno: ESNOMEM

Action: Determine the reason for lack of storage.

**Operation already in progress**

Reason: The user issues a call to the socket library of a type that the socket is trying to complete on a previous call.

errno: ESALREADY

Action: Wait, using `select()`, to determine when it is proper to issue another call of this type.

**Operation not supported on socket**

Reason: The user issues a socket function that is not supported by this type of socket. Cases of this error are issuing a `listen()`, `accept()`, or `connect()` function on a connectionless socket.

errno: ESOPNOTSUPP

Action: The socket user should determine the type of socket being used and verify that the proper functions are being issued to it.

**Operation now in progress**

Reason: A request issued to the socket library has been initiated by the library and the user can determine when it is complete by using the `select()` function call.

errno: ESINPROGRESS

Action: The user can issue a `select()` call to determine when the request has completed.

### Operation would block

Reason: An operation requested on a socket has the nonblocking I/O option set, this blocks the caller until some network-related event occurs.

errno: EWOULDBLOCK

Action: This is more a warning than an error. Issue the request at some later time. Depending on the request, the user can issue a `select()` function call to determine when this time is.

### Option not supported by protocol

Reason: The selected option for a `setsockopt()` or `getsockopt()` function call is not supported by the protocol layers. This message should not be generated because the error code is not currently set by the socket library.

errno: ESNOPROTOOPT

Action: Verify that the proper options are being passed to this function.

### Permission denied

Reason: The user does not have the proper authorization to make the request.

errno: EACCESS

Action: Determine the authorization to use the resource and then retry.

### Protocol family not supported

Reason: This error message is never generated because the error code is never set.

errno: EPROTONOSUPPORT

### Protocol not supported

Reason: The user issues a `socket()` and specifies a socket type that is not supported.

errno: EPROTONOSUPPORT

Action: Verify that the proper arguments are being passed to the `socket()` function.



**Protocol wrong type for socket**

Reason: The user issues a `socket()` function specifying both a socket type and a protocol to use. No support for a socket of this type is allowed.

errno: EPROTOTYPE

Action: Determine the correct settings for the arguments to the `socket()` function.

**Socket is already connected**

Reason: The user tries to issue a `connect()` function call on a socket that is already connected or to transfer data on a connected or associated socket and the user passes a name to send the data to.

errno: EISCONN

Action: Ensure that the user application is issuing the connect or send request on the proper socket.

**Socket is not connected**

Reason: The user tries to transfer data on a socket requiring connections but has not yet been connected.

errno: ENOTCONN

Action: First issue a `connect()` function call and then transfer data.

**Socket operation on non-socket**

Reason: The user issues a request on a file descriptor when the request can only be performed on a socket. This message should never be generated.

errno: ESOCK

Action: Verify that the proper file or socket descriptor is being used.

### Socket table overflow

Reason: The user currently has too many open sockets.

errno: ESNFILE

Action: Close some of the sockets or modify the socket configuration to allow more sockets per user.

### Socket type not supported

Reason: The error message should not be generated because the error code is never set.

errno: ESSOCKTNOSUPPORT

### Software caused connection abort

Reason: The connection was aborted by either the local or remote endpoint.

errno: ECONNABORTED

Action: If this error persists the user should verify the proper operation of Unicenter TCPaccess and the remote system being used.

### System Related Error

Reason: This error is generated if the API or Unicenter TCPaccess is stopped or terminated when a program using sockets is being executed. It can also be caused if the assembler API returns an error code to the socket library for which the socket library is not prepared.

errno: ESSYS

Action: First determine that the API and Unicenter TCPaccess are still functioning. If this is so and the error occurs repeatedly, contact Customer Support.

### Too many open sockets

Reason: The user currently has too many open sockets.

errno: ESMFILE

Action: Close some of the sockets or modify the socket configuration to allow more sockets per user.

**Transport provider ended**

Reason: The Unicenter TCPaccess subsystem was stopped by an operator command or has terminated abnormally.

errno: ESTPEND

Action: The socketterm() function should be called to close all open sockets.

**API ended**

Reason: The API was stopped by an operator command or has terminated abnormally.

errno: ESAPEND

Action: The socketterm() function should be called to close all open sockets.



# RPCINFO Messages

This chapter describes the messages issued by the RPCINFO program. It includes these sections:

- [Message Format](#) – Describes the standard format for messages issued by the RPCINFO program
- [RPCINFO Messages](#) – Lists all RPCINFO messages

## Message Format

Messages have the following format:

*RPC###t dow mon dd hh:mm:ss yyyy text*

<i>###</i>	Three-digit message number
<i>t</i>	One of these message types: <ul style="list-style-type: none"><li>I Informative</li><li>E Error</li><li>W Warning</li><li>D Debug</li></ul>
<i>dow</i>	Day of the week
<i>mon</i>	Month of the year
<i>dd</i>	Date
<i>hh</i>	Hours
<i>mm</i>	Minutes
<i>ss</i>	Seconds
<i>yyyy</i>	Year
<i>text</i>	message text

## RPCINFO Messages

This section lists the text of all RPCINFO messages in alphabetical order.

### 100I PROGRAM *d\_value1* VERSION *d\_value2* IS NOT AVAILABLE

Reason: RPCINFO finds that a particular RPC program is not available on a designated machine.

*d\_value1*      RPC program number

*d\_value2*      Version number of the RPC program in question

The difference between this message and message number 203 is that all communications worked with the portmapper—this message simply indicates that the RPC server is not executing on the selected machine.

### 101I PROGRAM *d\_value1* VERSION *d\_value2* is READY AND WAITING

Reason: The selected RPC server is running and ready for requests on the selected host.

*d\_value1*      RPC program number

*d\_value2*      Version number of the RPC program in question

### 102I NO REMOTE PROGRAMS REGISTERED

Reason: The selected host does not have any RPC servers registered with its local portmapper daemon.

### 103I COULD NOT DELETE REGISTRATION FOR PROGRAM *d\_value1* VERSION *d\_value2*

Reason: When trying to delete the registration of an RPC service with the local portmapper, the request was ignored by the portmapper because the service for the requested version of the requested was not currently registered.

*d\_value1*      RPC program number

*d\_value2*      Version number of the RPC program in question

**105I - 106I 105I number version protocol portname  
106I d\_value1 d\_value2 string1 d\_value3 string2**

Reason: RPCINFO requests a dump of a selected portmapper's registration tables.

<i>d_value1</i>	RPC program number
<i>d_value2</i>	Version number
<i>string1</i>	Either UDP or TCP, or the protocol number of the selected service
<i>d_value3</i>	Port number on which the service is listening for requests
<i>string2</i>	Either blank if the service is not known to the DNR, or else the name of the service as known by the DNR

**107I-110I 107I RPCINFO PROTO(protocol) HOST(hostname) PROG(prognum)  
{VERS(versnum)} {PORT(portnum)} {SYSID(subsystem\_id)}****108I RPCINFO HOST(hostname) {SYSID(subsystem\_id)}****110I RPCINFO DELETE PROG(prognum) VERS(vers\_num)  
{SYSID(subsystem\_id)}**

Reason: These messages are issued when the RPCINFO program is called with invalid parameters.

Action: Reissue the command with the proper parameter configuration.

**200E COULD NOT LOCATE LOCALHOST name**

Reason: The RPCINFO program could not locate the local host's name using DNR.

**201E UDP SOCKET**

Reason: RPCINFO could not open a UDP socket. This message is issued by perror() to provide more information as to the cause of the failure.

**202E PROGRAM d\_value IS NOT AVAILABLE**

Reason: RPCINFO finds that a particular RPC program is not available on a designated machine. The *d\_value* is the RPC program number.

**203E PROGRAM *d\_value1* VERSION *d\_value2* IS NOT AVAILABLE**

*d\_value1*      RPC program number

*d\_value2*      Version number of the RPC program in question

Reason:              RPCINFO finds that a particular RPC program is not available on a designated machine.

**204E CAN'T CONTACT PORTMAPPER**

Reason:              RPCINFO could not properly communicate with the selected host's portmapper.

Action:              Ensure that the selected host is running a portmapper daemon.

**207E *string* IS UNKNOWN RPC SERVICE**

Reason:              When trying to look up the name or program number of an RPCINFO program, the name or number could not be located. The *string* is the value of the requested entity.

**208E *string* IS UNKNOWN HOST**

Reason:              The DNR could not locate the Internet address for a host. The *string* is the name of the host.

**210E EXITING DUE TO PREVIOUS ERROR**

Reason:              When a fatal error is detected by RPCINFO, this error follows an error message that provides detailed information about the error. The RPCINFO program exits with a return code of 16 instead of zero (a normal exit return code).

**2xxE CAN'T CONTACT PORTMAPPER**

Reason:              RPCINFO could not properly communicate with the selected host's portmapper. This message is formatted by `clnt_pcreateerror()` before it is issued.

Action:              Ensure that the selected host is running a portmapper daemon.



**2xxE TCP CREATE**

Reason: RPCINFO could not create a TCP client RPC handle. The message is formatted by `clnt_pcreateerror()` before it is issued by the default `rpclog()` module. The message number 2xx is replaced with the sum of 250 plus the RPC library error number.

**2xxE UDP CREATE**

Reason: RPCINFO could not create a UDP client RPC handle. The message is formatted by `clnt_pcreateerror()` before it is issued by the default `rpclog()` module. The message number 2xx is replaced with the sum of 250 plus the RPC library error number.



This chapter provides reference information on Remote Procedure Call (RPC) error handling.

It includes:

- [RPC Library Error Messages](#) – Lists error messages generated by the RPC library
- [Message Formatting](#) – Describes the error message formatting routines `clnt_spcreateerror()`, `clnt_sperrno()`, and `clnt_sperror()`.RPC Log Interface

When an error is detected by the RPC library, it calls an externally defined function called `rpclog()`. The default `rpclog` shipped with the RPC library simply formats the information passed it and then print it to `stderr`.

## RPC Library Error Messages

This section lists error messages generated by the RPC library. Messages are listed in alphabetical order. For each message, the error number and csect string (csectp) are also given.

**Note:** Error Number 34 is not used.

### AUTHUNIX\_CREATE OUT OF MEMORY

Reason: Error Number 1 – Could not `malloc()`.  
csectp – AUTHUNIX\_CREATE

### CACHE\_SET COULD NOT ALLOCATE NEW RPC\_BUFFER

Reason: Error Number 33 – The `malloc()` function for RPC buffer failed.  
csectp – SVCUDP\_BUFCREATE

### **CACHE\_SET VICTIM NOT FOUND**

Reason: Error Number 31 – Corrupted cache caused search to be aborted.  
csectp – SVCUDP\_BUFCREATE

### **CACHE\_SET VICTIM ALLOC FAILED**

Reason: Error Number 32 – The malloc() function for cache element failed.  
csectp – SVCUDP\_BUFCREATE

### **CLNT\_BROADCAST BROADCAST DESERIALIZATION PROBLEM**

Reason: Error Number 9 – Received a bad RPC reply.  
csectp – CLNT\_BROADCAST

### **CLNT\_BROADCAST BROADCAST RPC NOT SUPPORTED**

Reason: Error Number 35 – Broadcast RPC is not supported.  
csectp – CLNT\_BROADCAST

### **CLNT\_BROADCAST BROADCAST SELECT PROBLEM**

Reason: Error Number 13 – The select() function failed when broadcasting.  
csectp – CLNT\_BROADCAST

### **CLNT\_BROADCAST CANNOT CREATE SOCKET FOR BROADCAST RPC**

Reason: Error Number 10 – The socket() routine failed.  
csectp – CLNT\_BROADCAST

### **CLNT\_BROADCAST CANNOT RECEIVE REPLY TO BROADCAST**

Reason: Error Number 14 – Client did not receive a reply to a broadcast request.  
csectp – CLNT\_BROADCAST

**CLNT\_BROADCAST CANNOT SEND BROADCAST PACKET**

Reason: Error Number 12 – The `sendto()` routine using broadcast address failed.  
csectp – CLNT\_BROADCAST

**CLNT\_BROADCAST CANNOT SET SOCKET OPTION SO\_BROADCAST**

Reason: Error Number 11 – The `setsockopt()` routine for `SO_BROADCAST` failed.  
csectp – CLNT\_BROADCAST

**CLNT\_PCREATEERROR string**

Reason: Error Number 3 – The string is replaced with a message generated by `clnt_screateerror()`. See [clnt\\_screateerror\(\)](#) for the formatting of string.  
csectp – CLNT\_PCREATEERROR

**CLNT\_PERRNO string**

Reason: Error Number 5 – The string is replaced with a message generated by `clnt_sperrno()`. See [clnt\\_sperrno\(\)](#) for the formatting of string.  
csectp – CLNT\_PERRNO

**CLNT\_PERROR string**

Reason: Error Number 4 – The string is replaced with a message generated by `clnt_sperror()`. See [clnt\\_sperror\(\)](#) for the formatting of string.  
csectp – CLNT\_PERROR

**CLNTRAW\_CREATE FATAL HEADER SERIALIZATION ERROR**

Reason: Error Number 6 – Could not XDR RPC call header.  
csectp – CLNTRAW\_CREATE

**CLNTTCP\_CREATE OUT OF MEMORY**

Reason: Error Number 1 – Could not `malloc()`.  
csectp – CLNTTCP\_CREATE

### **CLNTUDP\_CREATE OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – CLNTUDP\_CREATE

### **GET\_MYADDRESS GETHOSTNAME**

Reason: Error Number 7 – The gethostname() function failed.  
csectp – GET\_MYADDRESS

### **GET\_MYADDRESS GETHOSBYNAME**

Reason: Error Number 8 – The gethostbyname() function failed.  
csectp – GET\_MYADDRESS

### **MAKEFD\_XPRT OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – MAKEFD\_XPRT

### **MARSHAL\_NEW\_AUTH FATAL MARSHALLING PROBLEM**

Reason: Error Number 2 – Could not XDR authentication structures.  
csectp – AUTHUNIX\_CREATE

### **MVS\_SVC\_RUN SELECT FAILED**

Reason: Error Number 22 – The select() function failed.  
csectp – MVS\_SVC\_RUN

### **REGISTERRPC CAN'T REASSIGN PROCEDURE NUMBER 0**

Reason: Error Number 16 – Procedure number is 0 on call to registerrpc().  
csectp – REGISTERRPC

**REGISTERRPC COULDN'T CREATE AN RPC SERVER**

Reason: Error Number 17 – Couldn't create an RPC server.  
csectp – REGISTERRPC

**REGISTERRPC COULDN'T REGISTER PROG *d\_value1* VERS *d\_value2***

Reason: Error Number 18 – Could not register RPC server with portmapper. *d\_value1* is the RPC program number and *d\_value2* is the RPC program version. number.  
csectp – REGISTERRPC

**SVC\_RUN SELECT FAILED**

Reason: Error Number 22 – The select() function failed.  
csectp – SVC\_RUN

**SVCAUTH\_UNIX BAD AUTH\_LEN GID *d\_value1* STR *d\_value2* AUTH *d\_value3***

Reason: Error Number 15 – UNIX credentials are invalid. *d\_value1* is the UNIX group ID, *d\_value2* is the authentication string length and *d\_value3* is the authentication length.  
csectp – SVCAUTH\_UNIX

**SVCTCP\_CREATE CANNOT GETSOCKNAME OR LISTEN**

Reason: Error Number 24 – The getsockname() or listen() function failed.  
csectp – SVCTCP\_CREATE

**SVCTCP\_CREATE OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – SVCTCP\_CREATE

**SVCTCP\_CREATE TCP SOCKET CREATION PROBLEM**

Reason: Error Number 23 – Could not create TCP socket.  
csectp – SVCTCP\_CREATE

### **SVCUDP\_BUFCREATE CANNOT GETSOCKNAME**

Reason: Error Number 26 – The getsockname() function failed.  
csectp – SVCUDP\_BUFCREATE

### **SVCUDP\_BUFCREATE OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – SVCUDP\_BUFCREATE

### **SVCUDP\_BUFCREATE UDP SOCKET CREATION PROBLEM**

Reason: Error Number 25 – Could not create UDP socket.  
csectp – SVCUDP\_BUFCREATE

### **SVCUDP\_CREATE OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – SVCUDP\_CREATE

### **SVCUDP\_ENABLECACHE CACHE ALREADY ENABLED**

Reason: Error Number 27 – Cache enable request when cache already enabled  
csectp – SVCUDP\_BUFCREATE

### **SVCUDP\_ENABLECACHE COULD NOT ALLOCATE CACHE**

Reason: Error Number 28 – The malloc() function for cache control space failed.  
csectp – SVCUDP\_BUFCREATE

### **SVCUDP\_ENABLECACHE COULD NOT ALLOCATE CACHE DATA**

Reason: Error Number 29 – The malloc() function for cache data space failed.  
csectp – SVCUDP\_BUFCREATE



**SVCUDP\_ENABLECACHE COULD NOT ALLOCATE CACHE FIFO**

Reason: Error Number 30 – The malloc() function for cache FIFO failed.  
csectp – SVCUDP\_BUFCREATE

**UNIVERSAL COULD NOT SEND REPLY**

Reason: Error Number 19 – Could not send reply.  
csectp – REGISTERRPC

**UNIVERSAL NEVER REGISTERED PROG *d\_value***

Reason: Error Number 21 – Program *d\_value* was never registered. *d\_value* is the RPC program number.  
csectp – REGISTERRPC

**UNIVERSAL TROUBLE REPLYING TO PROG *d\_value***

Reason: Error Number 20 – Could not send reply. *d\_value* is the RPC program number.  
csectp – REGISTERRPC

**XDR\_ARRAY OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – XDR\_ARRAY

**XDR\_BYTES OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – XDR\_BYTES

**XDR\_RECORD OUT OF MEMORY**

Reason: Error Number 1 – Could not malloc().  
csectp – XDR\_RECORD

## XDR\_REFERENCE OUT OF MEMORY

Reason: Error Number 1 – Could not malloc().  
csectp – XDR\_REFERENCE

## XDR\_STRING OUT OF MEMORY

Reason: Error Number 1 – Could not malloc().  
csectp – XDR\_STRING

# Message Formatting

The following error message formatting routines are described in this section:

- clnt\_screateerror()
- clnt\_sperrno()
- clnt\_sperror()

## clnt\_screateerror()

Reason: This routine formats a message about errors related to the creation of a client handle. It should be passed a pointer to user message text. It formats the message this way and then returns a pointer to the formatted message:

%s: %ns\n

%s User-supplied message text.

%ns Message text generated by clnt\_sperrno() acting on the client status.

If the creation error is neither RPC\_PMAPFAILURE nor RPC\_SYSTEMERROR, a new line is appended. Otherwise these errors are additionally added:

- For RPC\_PMAPFAILURE:

%s: %ns - %es\n

%s User supplied message text.

%ns Message text generated by clnt\_sperrno() acting on the client status.

%es Message text generated by lnt\_sperrno() acting on the RPC error status.

■ For `RPC_SYSTEMERROR`:

`%s: %ns - %ss\n`

`%s`      User supplied message text

`%ns`      Message text generated by `clnt_sperrno()` acting on the client status.

`%ss`      Message text generated by indexing into the `perror()` socket library error table using the RPC err.

If the error is not in the socket library `perror()` error table `%ss` is replaced with `ERROR %d` where `%d` is the RPC library `errno` value.

## `clnt_sperrno()`

Reason: This routine generates a message about an RPC library error. This routine should be passed an enum `clnt_stat` argument. A pointer to the message text relating to the `clnt_stat` argument is returned.

It can be one of the following:

RPC: SUCCESS	Successful completion
RPC: CAN'T ENCODE ARGUMENTS	Client could not XDR the arguments it is to pass to the remote procedure.
RPC: CAN'T DECODE RESULT	Client could not XDR the result returned from the remote procedure.
RPC: UNABLE TO SEND	Client could not send an RPC CALL to the remote procedure.
RPC: UNABLE TO RECEIVE	Client could not receive the RPC REPLY from the remote procedure.
RPC: TIMED OUT	Client did not get a response from the server within an allowable amount of time.
RPC: INCOMPATIBLE VERSIONS OF RPC	The versions of the RPC protocol used by the server and the client are not equal.
RPC: AUTHENTICATION ERROR	Authentication check failed on remote system.

RPC: PROGRAM UNAVAILABLE	Program is not available on remote system.
RPC: PROGRAM/VERSION MISMATCH	Program and version are not available on the remote system.
RPC: PROCEDURE UNAVAILABLE	Requested procedure of the selected program on the remote system is not available for use.
RPC: SERVER CAN'T DECODE ARGUMENTS	Remote system could not understand arguments passed to the selected program.
RPC: REMOTE SYSTEM ERROR	Remote system had a major failure trying to execute the selected program.
RPC: UNKNOWN HOST	User selected a remote host system that is unknown to the DNR.
RPC: UNKNOWN PROTOCOL	User specified an unsupported protocol to be used for transport.
RPC: PORT MAPPER FAILURE	Remote host's portmapper could not be communicated with properly.
RPC: PROGRAM NOT REGISTERED	Remote program is not registered with the remote host's portmapper.
RPC: FAILED (UNSPECIFIED ERROR)	Error was not specific enough to justify its own error code.
RPC: (UNKNOWN ERROR CODE)	Error is not decipherable.

## clnt\_sperror()

Reason:

This error message formatting routine requires two arguments:

- A pointer to a client handle
- A pointer to user message text

It formats a message and returns a pointer to the formatted message text.

The user message is formatted followed by a colon and a space (for example, "%s: "). A message generated by `clnt_sperrno()` (see [clnt\\_sperrno\(\)](#)) follows; the message formatting varies based on the current error status.

If the error is any of these, a new line is attached to the user portion and the pointer to the text returned:

- `RPC_SUCCES`
- `RPC_CANTENCODEARGS`
- `RPC_CANTDECODERES`
- `RPC_TIMEDOUT`
- `RPC_PROGUNAVAIL`
- `RPC_PROCUNAVAIL`
- `RPC_CANTDECODEARGS`
- `RPC_SYSTEMERROR`
- `RPC_UNKNOWNHOST`
- `RPC_UNKNOWNPROTO`
- `RPC_PMAPFAILURE`
- `RPC_PROGNOTREGISTERED`
- `RPC_FAILED`

Thus, the message for these looks like this:

```
s: %ns\n
```

An example is:

```
clnt_sperror(clntp, "USER MESSAGE");
```

The `clnt_sperrno` routine returns

```
RPC: CAN'T DECODE RESULT t
```

The message pointer points to

```
USER MESSAGE: RPC: CAN'T DECODE RESULT\n
```

If the error is `RPC_CANTSEND` or `RPC_CANTRECV`, the user portion of the message is followed by an entry from the `perror()` error message list of the socket library.

The generalized format of the message looks like this:

```
%s: %ns ; errno = %ss\n
```

%s        User message passed.  
 %ns        clnt\_sperrno() generated message text.  
 %ss        Message from the socket library.

If the error status is `RPC_VERSIONMISMATCH`, the message is formatted like this:

```
%s: %ns ; LOW VERSION = %lul, HIGH VERSION = %luh\n
```

%s        User message passed.  
 %ns        clnt\_sperrno() generated message text.  
 %lul       Lowest decimal version number of the RPC program running.  
 %luh       Highest decimal version number of the RPC program running.

If the error status is `RPC_AUTHERROR`, the message is formatted like this:

```
%s: %ns ; why = %sa:
```

%s        User message provided.  
 %ns        clnt\_sperrno() generated message text.  
 %sa        Authentication message generated.

These are the authentication messages:

Message	Description
AUTHENTICATION OK	The authentication is OK.
INVALID CLIENT CREDENTIAL	Client authentication credentials are incorrect for authentication type.
SERVER REJECTED CREDENTIAL	Client credentials do not allow access to the procedure.
INVALID CLIENT VERIFIER	Credentials are not supported by client.
SERVER REJECTED VERIFIER	Server could not decode this type of authentication.

Message	Description
CLIENT CREDENTIAL TOO WEAK	Client credentials formatted properly but are of too low authentication to allow access.
INVALID SERVER VERIFIER	Credentials are not supported by server.
FAILED (UNSPECIFIED ERROR)	Authentication failed for an error that does not justify a more specific error code.
UNKNOWN AUTHENTICATION ERROR - %d	The authentication error is unknown to the RPC library. The %d is replaced with the decimal value of the authentication error.

If the error status is `RPC_PROGVERSMISMATCH`, the message is formatted this way:

`%s: %ns ; LOW VERSION = %lu1, HIGH VERSION = %lu2\n`

`%s`      User message provided.  
`%ns`      `clnt_sperrno()` generated message text.  
`%lu1`     Lowest version of the RPC program running.  
`%lu2`     Highest version of the RPC program running.

If the error status is not mentioned, the message is formatted this way:

`%s: %ns ; S1 = %lu1, S2 = %lu2\n`

`%s`      User message provided.  
`%ns`      `clnt_sperrno()` generated message text.  
`%lu1`     First error argument.  
`%lu2`     Second error argument.





This chapter describes unnumbered messages issued to the client by Server Telnet.

## Server Telnet Messages

This section includes the messages that are issued by Server Telnet.

### Command Not Supported

Reason: There is no support for the command name entered, or the support is not properly defined in the configuration.

Action: Enter the proper command name, or consult with your system administrator to resolve the issue.

### Enter Command or HELP

Reason: This is the main user prompt issued by Server Telnet.

Action: In response, enter the name for a command, service, or VTAM application supported by the Server Telnet. Alternatively, enter HELP, for guidance.

### Operator Forced Logout

Reason: The TCP/IP provider has terminated, either abnormally or normally, in response to an MVS operator STOP command.

Action: Await restart of the TCP/IP provider, and then resume working.

### Password Not Authorized

Reason: The password entered does not match the user ID entered.

Action: Use a proper combination of user ID and password.

### Requested Command Cannot Be Serviced

Reason: The requested service could not be executed because its supporting module could not be found or loaded correctly.

Action: Report the matter to your system administrator for resolution.

### Requested Service Restricted

Reason: This message reports that the requested service is restricted and access will depend on your user ID and password.

Action: In response to subsequent prompts, enter your user ID and password.

### *prodid* Server Telnet (*hostid*) *timestamp*

Reason: This is the sign on banner for Server Telnet running under Unicenter TCPaccess. *prodid* identifies the version of Unicenter TCPaccess to which you are connected. *hostid* is the domain name (host name on the TCP/IP network) for this instance of Unicenter TCPaccess. *timestamp* reports the date and time the connection was established.

### Server Telnet User *uid* Logged Out

Reason: This message reports that the user ID *uid* has been logged out. This message appears after a service has been executed or a change of user ID is changing.

### Service Completed Abnormally

Reason: The module providing the service terminated abnormally.

Action: Report the failure to your system administrator for resolution.

**Translate Table Cannot Be Loaded**

- Reason: The translate table specified by the TRANTBL parameter on the TELNET or APPL statement cannot be found or cannot be loaded properly. Consequently the service requested could not be executed.
- Action: Have your system administrator correct the configuration or make available the proper translate table load module.

**Userid is Not Defined To the Security System**

- Reason: An unknown user ID was entered for access to a restricted service.
- Action: Enter a user ID and password that is authorized for the service.







Computer Associates™